Augmenting People in Monocular Video Data

Von der Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines
Doktoringenieurs (Dr.–Ing.)

genehmigte Dissertation

von
Lorenz Rogge
geboren am 06.01.1984
in Wernigerode

Eingereicht am:	18.05.2015
Disputation am:	03.07.2015
1. Referent:	Prof. Dr.–Ing. Marcus Magnor
2. Referent:	Prof. Dr.–Ing. Bodo Rosenhahn

(2015)

Abstract

When aiming at realistic video augmentation, i.e. the embedding of virtual, 3-dimensional objects into a scene's original content, a series of challenging problems has to be solved. This is especially the case when working with solely monocular input material, as important additional 3D information is missing and has to be recovered during the process, if necessary. In this work, I will present a semi-automatic strategy to tackle this task by providing solutions to individual problems in the context of virtual clothing as an example for realistic video augmentation. Starting with two different approaches for monocular pose and motion estimation, I will show how to build a 3D human body model by estimating detailed shape information as well as basic surface material properties. This information allows to further extract a dynamic illumination model from the provided input material. The illumination model is particularly important for rendering a realistic virtual object and adds a lot of realism to the final video augmentation. The animated human model is able to interact with virtual 3D objects and is used in the context of virtual clothing to animate simulated garments. To achieve the desired realism, I present an additional image-based compositing approach that realistically embeds the simulated garment into the original scene content. Combining the presented approaches provide an integrated strategy for realistic augmentation of actors in monocular video sequences.

Zusammenfassung

Unter der Zielsetzung einer realistischen Videoaugmentierung durch das Einbetten virtueller, dreidimensionaler Objekte in eine bestehende Videoaufnahme, gibt eine Reihe interessanter und schwieriger Problemen zu lösen. Besonders im Hinblick auf die Verarbeitung monokularer Eingabedaten fehlen wichtige räumliche Informationen, welche aus den zweidimensionalen Eingabedaten rekonstruiert werden müssen. In dieser Arbeit präsentiere ich eine halbautomatische Verfahrensweise, welche es ermöglicht, die einzelnen Teilprobleme einer umfassenden Videoaugmentierung nacheinander in einer integrierten Strategie zu lösen. Dies demonstriere ich am Beispiel von virtueller Kleidung. Beginnend mit zwei unterschiedlichen Ansätzen zur Posen- und Bewegungsrekonstruktion wird ein realistisches 3D Körpermodell eines Menschen erzeugt. Dazu wird die detaillierte Körperform durch ein geeignetes Verfahren approximiert und eine Rekonstruktion der Oberflächenmaterialen vorgenommen. Diese Informationen werden unter anderem dazu verwendet, aus dem Eingabevideo eine dynamische Szenenbeleuchtung zu rekonstruieren. Die Beleuchtungsinformationen sind besonders wichtig für eine realistische Videoaugmentierung, da gerade eine korrekte Beleuchtung den Realitätsgrad des virtuell generierten Objektes erhöht. Das rekonstruierte und animierte Körpermodell ist durch seinen Detailgrad in der Lage, mit virtuellen Objekten zu interagieren. Dies kommt besonders im Anwendungsfall von virtueller Kleidung zum tragen. Um den gewünschten Realitätsgrad zu erreichen, führe ich ein zusätzliches, bild-basiertes Korrekturverfahren ein, welches hilft, die finale Bildkomposition zu optimieren. Die Kombination aller präsentierter Teilverfahren bildet eine vollumfängliche Strategie zur Augmentierung von

monokularem Videomaterial, die zur realistischen Simulation und Einbettung von virtueller Kleidung eines Schauspielers im Originalvideo verwendet werden kann.

Acknowledgments

After many great years of work and research at the *Institut für Computergraphik* (ICG) there are a lot of people I want to express my gratitude to. First of all I want to thank my parents Vera and Jörg who always supported my ideas and decisions in life. They always encouraged me to follow my interests which lead me to starting computer science studies at TU Braunschweig and, eventually, becoming part of the team at ICG as a graduate student. I also want to thank my fiance Tina for her loving support even during hard weeks of work, especially during several paper deadlines and during writing this thesis.

Great gratitude goes to my supervisor Marcus Magnor, who gave me the opportunity to work at *Institut für Computergraphik* first as a student assistant and later as a graduate student. With his great advice, support, and encouragement, I was able to work on many interesting topics in a great team of researchers and I was able to be part of the research community. I also thank him for giving me the opportunity to write my diploma thesis during a three-months visit at *University of New Mexico* in Albuquerque back in 2009.

Besides, I want to thank Christian Linz for being a great supervisor of my diploma thesis and for giving me first insights on the work at ICG together with Christian Lipski. Joined by Martin Eisemann, Georgia Albuquerque, Timo Stich, Kai Berger, Anita Sellent, Benjamin Meyer, Anja Franzmeier, Florian Barucha, and Carsten Götze they made me feel at home right from the start. Furthermore, I want to thank Thomas Neumann for several months of great collaboration on my first big project. Gratitude goes to Felix Klose and Michael Stengel for their intense support on the experiments of my work on video augmentation,

leading to an international journal publication, and for many detailed scientific discussions on various topics. Of course, I also want to thank Stephan Wenger for sharing an office for five years without complaints. On the contrary, we had a lot of fun and always shared many great ideas and thoughts. Furthermore, I want to thank Kai Ruhl, Maryam Mustafa, Benjamin Hell, Stefan John, and Pablo Bauszat for many helpful scientific discussions. Thanks to Anja Franzmeier, Kristina Branz, Florian Barucha and Ariana Prekazi for doing a great job in the secretary's office and thanks to Carsten Götze for keeping the computers running.

The research leading to this publication was also funded by the *European Union's Seventh Framework Programme FP7/2007-2013* under grant agreement no. *256941, Reality CG.*

Contributions of the Author

In the following I clarify my individual contributions to the publications that describe parts of my thesis. The individual publications are ordered according to the structure of this thesis.

- Lorenz Rogge, Thomas Neumann, Markus Wacker, and Marcus Magnor. **Monocular Pose Reconstruction for an Augmented Reality Clothing System**. In Proceedings of Vision, Modeling and Visualization (VMV), pages 339-346, September 2011.

  This paper resulted from a cooperation of Institut für Computergraphik *at* TU Braunschweig*,* Institut für Datentechnik *at* TU Braunschweig*, and* Drematrix *computer vision lab at* HTW Dresden*. The research leading to the publication's results was funded by the* European Union's Seventh Framework Programme FP7/2007-2013 *under grant agreement no.* 256941, Reality CG *and has also received funding in parts by the* German Science Foundation DFG, grant no. MA 2555/8-1*. My contributions to the paper were the development of the pose descriptor* Feature Context *and its combined use together with HOG descriptors in a robust pose reconstruction. Together with Thomas Neumann I discussed various ideas for feature descriptors and pose regression techniques. Being offered to use the motion capture stage at the HTW in Dresden, I was able to create high-quality training data with the help of Thomas Neumann. I was also responsible for the pipeline formulation and implementation of the pose reconstruction scheme as individual image processing steps. The technique was required to consist of only small separate processing blocks as it was implemented as a*

demonstrator for image processing applications on programmable hardware by Institut für Datentechnik. *While I received helpful insights to designing algorithms for programmable hardware from Daniel Thiele and Rolf Ernst, Marcus Magnor and Markus Wacker guided the image-processing side of the project with many suggestions and helpful advice regarding ideas for feature detection and pose regression. The contributions to this paper are part of Chapter 3 of this thesis.*

- Lorenz Rogge, Felix Klose, Michael Stengel, Martin Eisemann, and Marcus Magnor. **Garment Replacement in Monocular Video Sequences**. In ACM Transactions on Graphics (ACM TOG), vol. 34, no. 1, pages 6:1-6:10, November 2014.

  The research leading to this publication was also funded by the European Union's Seventh Framework Programme FP7/2007-2013 *under grant agreement no.* 256941, Reality CG. *My contributions to this publication were the design of a video augmentation technique as an integrated strategy of consecutive image processing steps. While some parts like actor segmentation were solved using existing approaches, I developed and implemented a novel and robust strategy to estimate pose and shape of an actor using a parameterized body model, multi-staged parameter optimization, and an underlying key-frame structure allowing for optional manual corrections. I also developed and implemented a novel approach to geometry aided reconstruction of dynamic scene illumination from monocular video. With the great support of Michael Stengel I was able to create highly realistic garment simulations, for which he designed the virtual garments and mate-*

rials used for rendering. Together with Felix Klose I discussed major parts of the image-based motion correction technique and he also helped with the implementation in this field during the experiments. Martin Eisemann, Felix Klose and Michael Stengel also helped writing parts of the paper, while Marcus Magnor supervised the project and provided helpful advice and ideas. The contributions to this publication are part of Chapter 4, Chapter 5, and Chapter 6 of my thesis.

- Lorenz Rogge, Pablo Bauszat, and Marcus Magnor. **Monocular Albedo Reconstruction**. In Proceedings of IEEE International Conference on Image Processing (ICIP), pages 1046-1050, October 2014.
  This publication is based on the work on video augmentation of my ACM TOG paper and contains a more sophisticated approach to albedo reconstruction from monocular video data. Being part of the project Reality CG *the research on this topic was funded by the* European Union's Seventh Framework Programme FP7/2007-2013 *grant aggreement no.* 256941. *My contributions to this publication were the development and implementation of the monocular albedo reconstruction based on statistical color analysis. Together with Pablo Bauszat I discussed the benefits of ambient occlusion information for better albedo reconstruction which I then integrated into the color sampling as a weighting factor. The contributions to the project are part of Chapter 5, Section 5.2. Marcus Magnor supervised the project with helful suggestions and advice.*

Contents

1 Introduction

The augmentation of video material allows integrating additional information or virtual objects into an existing scene. This can be useful in video and movie production, as recorded scenes can be improved and altered in an augmentation post-process. As modern movie productions already heavily rely on computer generated imagery (CGI), composing real actors into artificially generated environments [Fos14] or replacing actors with computer-generated and animated avatars [Gel08], it seems promising and straight forward to use image-based video augmentation techniques to improve and augment objects or actors in recorded video material with virtual objects instead of replacing them. The goal of this thesis is, therefore, to analyze and develop a strategy to augment existing monocular video data in a realistic way.

There are image-based video augmentation systems available, that allow to augment monocular video streams with information or additional objects. For example, systems augmenting video with simple information can be integrated in portable devices, such as *Google Glass* [Goo] or mobile apps on smart phones [Meta]. Applications on these devices analyze 2D scene content and structures to retrieve information about the object of interest. For predefined patterns and objects, additional, textual information can be superimposed over the original

video. It is also possible to derive an object's orientation or its surface properties in 3D space and align additional 3D geometry appropriately.

Metaio uses this technique for marketing purposes. For example in one of their applications used by *IKEA*, a 3D visualization of furniture and the projection into the real world allows to virtually *try out* the furniture in your own home [Metb]. They use the *IKEA* catalog itself as a reference marker and project a selected 3D furniture model into the original input data by aligning it to the reference marker.

Video augmentation has also been used for games for a rather long time. With the more common use of cameras in addition to game consoles, the use of video data as an input device in terms of game control but also for direct augmentation purposes got a significant boost with the start of the *Eye Toy* camera of Sony's *PlayStation 2* [PBBDN09]. The original input video is augmented with game assets and the user is able to interact with these assets using a basic motion tracking technique. More sophisticated applications try to use the camera as a marker tracking device in order to augment the input video with more realistic 3D graphics that are projected into the original scene more correctly. The game *Eye of Judgment* of *Sony Computer Entertainment*, for example, augments an originally flat board game with 3-dimensional board game playing figures [Son]. The individual cards representing the game characters in the real world serve as 2D markers identified by unique codes. The system tracks and identifies these markers in the video stream and derives a 3D projection from their position and distortion in camera space. Finally, a virtual 3D model of each identified game figure is realistically projected into the video, such that a 2D board game is augmented towards a 3D board game on TV. With this augmentation, the 3D

board game is more dynamic compared to an actual 3D board game equivalent, as the digital augmentation allows for animated game assets and additional visual effects. This example clearly shows the benefits of video augmentation in consumer media. It allows for additional game play assets that can be interacted with, but could have not been realized in the real world game. Still, the virtual objects are not used to realistically augment existing objects or actors, as they just replace objects or are used as overlays.

In addition to the detection of flat, two-dimensional markers and estimating their orientation in space, deformable surfaces can be tracked and used for augmentation of flexible and deformable objects, such as fabrics and clothing. By allowing the coded marker to deform locally, it is still possible to identify and track the 2D manifold described by the marker. This can be used to track and augment clothes and fabrics in video streams [SM06].

A promising application for this kind of video augmentation is a virtual mirror experience [EFR08] or the augmentation of actors in a video with artificial garments [HE09]. In case of a virtual mirror, a person steps in front of a combination of screen and camera, instead of a real mirror. The camera is then be used to capture the person, while an image processing device is used for analyzing the video stream, identifying and tracking patterns on the person's surface or even the actual body motion. In an image-based augmentation process, virtual clothing can be generated and superimposed over the original video data, which is then be displayed on the screen. This achieves the illusion of a mirrored image of the person wearing virtual apparel. Interactions of the virtual garment and the real person can be realized by analyzing and reconstructing the original scene's content and the person's animation and body properties at a high level

of detail. A reconstructed and realistically shaped avatar can then be used to interact with virtual objects in a simulation.

In times of increasing online shopping and also in the field of apparel design, a technique like this could be very relevant for virtual try-on systems [KF08; WKK+04]. To speed up the process of garment design and to improve rapid prototyping, a virtual cloth augmentation technique can be used to design and test first prototypes completely virtually without any need of fabricating each individual garment. The different prototypes can be tested in digital simulations using motion data of real models providing a realistic animation. The removal of the production step of these design prototypes speeds up the development process and saves money. For online shopping, a virtual try-on of clothes is also very promising. Customers can virtually try on the clothes at home prior to placing an order. This would also effectively reduce the number and cost of returned orders to the online retailer.

The realistic augmentation of actors in existing video material is also be beneficial for movie productions. After capturing a scene with real actors, the clothing and accessories cannot be easily exchanged. If the director decides, different clothing would fit better into a current setting, the scene has to be captured again. The solution would be to either record the scene again, this time with the correct clothes and accessories, or to manually edit the material frame by frame using properly aligned and synchronized CGI. Both solutions are costly and tedious and could be significantly improved by using a semi-automatic augmentation approach. The reconstruction of a highly realistic actor model and its motion, the analysis of scene illumination, and a realistic image-based augmentation help to reduce the modeling and compositing labor of digital

artists, while completely removing the need of an additional recording of the scene. Eventually, this would reduce production time and cost of movie or video productions.

To achieve the goal of a high quality video augmentation, sophisticated scene analysis and actor reconstruction is required in combination with high quality rendering techniques. The realization of a virtual mirror additionally requires the augmentation to run at real-time frame rates to allow for an interactive experience, while in the application of actor augmentation as a video post-process, the focus moves more towards realism, to achieve highly realistic and believable augmentations of the original video. As a post-process, a detailed analysis of the original video data and the generation of highly realistic models of the object of interest and the environment are possible. Also physically correct simulation and rendering of the artificial objects is possible, improving the desired realism even more.

In the following I will present two different approaches towards the augmentation of an actor in monocular video material. The first approach focuses on a fast and robust motion reconstruction technique that can be used in interactive augmentation applications. Moving the requirement away from interactive computation times towards a highly realistic augmentation, in a second approach I propose a strategy to reconstruct a highly detailed actor model and a realistic illumination model from a monocular input sequence. These models can then be used to realistically augment the original actor with artificial clothing.

2 Related Work

The approaches I propose to solve the individual problems encountered on the way towards realistic video augmentation combine and extend techniques from various fields of research. Starting with image analysis regarding detection, extraction and detailed description of humans in images, the realistic reconstruction of three-dimensional objects is of interest. Furthermore, a scene's properties such as global illumination and camera motion are important. Lastly, for the final augmentation application, a realistic rendering technique and video compositing approaches are necessary to create believable and realistic video augmentations.

2.1 Detecting People using Feature Analysis

When it comes to augmenting people in video data, the first task to solve is detecting humans in the given input data. For some applications, it may already be sufficient to detect the coarse location of a person, without extracting more detailed information. A given image has to be analyzed regarding certain visible parameters to decide whether there is a person and where he/she is. As people wear various styles of clothing, it is not easy to distinguish between a person or

any other regular object just by looking at its color and appearance. A possible approach would be to find and process skin-colored regions of the image to detect a person's head and hands [JR02], as these body parts are seldomly covered by clothing or other objects. Given a plausible configuration of these detected regions, a human can be considered as detected [CL04].

However, for object detection in commodity surveillance systems and similar applications based on cheap monochrome or infrared cameras, color images are usually not available. Therefore, approaches based on (skin)-color classification are not feasible. Motion-based detection systems allow for robust detection of objects or obstacles, e.g., in the field of automotive assistance systems [SCF09]. These approaches aim at detecting objects, in general, but still lack detail in the detection of specific poses or the classification of object features and their location.

Using Haar-like feature descriptors, regions can be detected where an object or a person is most likely to be found [VJ01]. These feature descriptors analyze prominent edge configurations typical for humans, such as limb positions and the shape of the head. A certain combination of Haar-Wavelet patterns, representing a human shape, can be used for filtering the image yielding probabilities for every detector part. The combined likelihood of all patterns serves as overall likelihood of a person. A detector using this technique is able to detect and locate humans in given images just by applying 2D image filter operations. Thus, they are very fast and feasible for real-time applications [GS08; YD12]. In subsequent research, these descriptors have been extended to detect sub-parts of people, such as upper torso regions, left and right body parts or the legs. These

kinds of descriptors are able to detect and locate people in images, even if they are partly occluded by other people or objects in the scene [EG09; LSS05].

2.2 Monocular Silhouette Segmentation

The detection of human actors in image data is not sufficient for the processing steps required for video augmentation. For analysis and evaluation of the body's configuration and pose, the outline of the person is important as it provides many cues about the location of certain body parts as well as the overall body shape and even the location and type of worn clothing. Besides manual segmentation realized with professional rotoscoping tools, e.g. *Nuke* [The], as most commonly used for commercial movie productions, automated approaches have been developed to extract a human silhouette from monocular input images of videos. Some approaches still require manual initialization, as this provides information about the color, material, and shape properties of the object to be segmented [BWSS09; RKB04]. However, based on a-priori knowledge about the objects of interest or motion information, these initialization steps can be omitted, and models for color and shape can be updated and evaluated on-the-fly allowing for real-time video segmentation [CCBK06].

2.3 Pose-Descriptors

Detected features allow one to distinguish between certain body parts or to track them over time to identify motion elements. However, analyzing the

combination of a set of features allows one to describe a person's appearance as a whole.

Using information about the silhouette edge or edge orientations of a person, a pose-specific descriptor can be derived that is able to describe an observed pose and distinguish even between similar poses. The *Shape Context* descriptor uses the silhouette edge to identify objects [BMP02] or describe an object's pose configuration [MM02]. By sampling the edge distances of silhouette edge samples to a central silhouette point and accumulating these sample values in a log-polar histogram allows one to eventually describe the silhouette's shape as a whole by the shape of the histogram alone, Figure 2.1. This histogram is called *Shape Context* [BMP00].

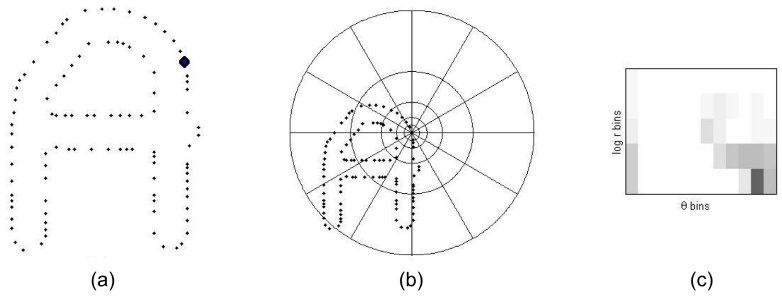

Figure 2.1: *Shape Context* Silhouette Descriptor [BMP00]: In the *Shape Context* scheme the object's contour is regularly sampled (a) and for every boundary point, the neighboring boundary points are evaluated regarding their orientation and distance (b) which are accumulated in a log-polar histogram (c). This histogram allows comparing and distinguishing various contour shapes by analyzing the local neighborhood of boundary points.

More elaborate analysis of the edge information in an image allows to even describe the pose of a person. Using histograms of oriented gradients (HOG)

it is possible to describe a human pose as a whole [DT05]. The region around the person is subdivided into a predefined set ob sub-regions. In all of these sub-regions the oriented gradient and its magnitude of all pixels, derived from the per-pixel derivative along the x- and y-axis, is accumulated. Finally the set of oriented gradients serves as a feature vector which is able to describe the pose of the person visible in the input data.

2.4 Machine Learning for Pose Regression

Using the silhouette or pose descriptors described above Section 2.3, it is possible to distinguish between different poses. Using machine learning, a system can be trained to estimate or classify the most plausible pose to a given feature vector [Gun98].

Using a Support Vector Machine *SVM* and training data, consisting of pairs of silhouette descriptors and 3D pose data, for every silhouette descriptor a corresponding pose angle vector can be generated [HSW98]. The SVM returns the best matching pose vector for a given pose descriptor. However, as the support vector machine is classifying the input descriptor to a predefined and fixed set of training samples, the result is always a discrete pose from this training data set. Smooth and contiguous pose or motion reconstruction is not possible using this approach. This problem can be solved by using Relevance Vector Machines *RVM* which yield interpolated results from several classified training data pose vectors [Tip00], such that smoother and spatially and temporally more consistent poses can be generated [AT04]. Finally, using a combination of

different silhouette descriptor techniques, a more stable pose can be reconstructed from single images or video data [AT06].

Another approach to pose estimation was presented by Stoll *et al.*who fitted 3D Gaussians in the multiview silhouette of a person and used the size and distribution of the Gaussians as a fast and efficient pose descriptor [SHG+11]. However, this descriptor requires multiview video data and is not able to handle monocular input data.

2.5 Parameterized Body Models

As it is not always sufficient to only locate a person and describe his/her pose, techniques had to be developed to describe a person's body shape as well. Using real-world laser scan data of numerous actors, varying in gender, age, body height and weight, it was possible to generate the parameterized model *SCAPE* describing the average human [ASK+05]. By fitting a standardized mesh model to all laser scan data, represented by point clouds, the meshes for different people could be compared properly as a direct mapping of surface regions was possible. Analyzing the per-vertex deformation between different actors allows to describe a vertex deformation vector for each variation of shape parameters. From there, a mapping of shape parameters to the mesh deformation could be derived using principal component analysis *PCA*, yielding a parameterized body mesh model. This model can be used to adapt the shape of a person by tuning its parameters to an optimum.

As the model created from straight-forward PCA does not provide any semantic relation of the shape parameters, Hasler *et al.*extended the model to support

such a semantic parameter mapping [HSS+09]. Using annotated parameter configurations of SCAPE models, a mapping from semantically meaningful parameters such as gender, body height, weight, or fitness was possible to the parameter set of the SCAPE model.

Such a model can be used to fit the body pose and shape to multiview data of an actor [HRT+09] by minimizing the projection error of the model to the actor's silhouette for all view points [CTMS03].

An adaptation to monocular image data was presented by Jain *et al.*who used the body model's parameterization and energy minimization to optimally fit shape and pose to a 2D silhouette [JTST10]. Tracking of surface features of the actor in a monocular video sequence allowed for a plausible motion reconstruction. By manually altering the shape parameters of the reconstructed body model, the original input image data could be distorted according to the visual shape difference of reconstructed and manually modified body model. This allows to change the actor's shape in the original video.

2.6 Object BRDF and Illumination Estimation

The task of augmenting an existing video with artificial content requires one to be able to render this virtual content as realistically as possible. Therefore, reconstructing pose, motion and shape of the person in the video does not suffice. As additional information, knowledge about the scene lighting is very important as this introduces a large amount of realism to the rendered content. To reconstruct the illumination from a given video sequence, without any knowledge about scene geometry is a challenging task. Given reference geometry and an

object's texture allows separating the illumination effects from the rest of the input data [Deb98]. The separated shading information then allows estimating the original scene lighting configuration [CWJ11]. This is achieved by inverting the photometric stereo approach, where 3D geometry is reconstructed from a set of given input images of the same object under different controlled illumination conditions. The user-controlled illumination allows one to separate surface material appearance from illumination effects, and using multiple illumination directions allows to estimate a possible surface normal for every surface point / pixel in the input images. In an energy minimization scheme, these surface normal candidates are optimized to a smooth surface solution, and a 3D geometry can be derived by integrating the estimated normal directions. Therefore, it is possible to estimate geometry from input images under known illumination conditions. If, in addition to the input images, the geometry is known, it is possible to invert the photometric stereo approach to estimate the directional vectors of incident light instead. However, since the illumination and material properties cannot directly be decoupled from a given input image, the object's surface properties must be given in addition to the geometry information. Given this data, it is possible to reconstruct the environmental illumination configuration [CWJ11; FKGK05].

Usually, from a given input video sequence, the original, unshaded object material cannot directly be inferred. Using statistical analysis [RBM14] or coupled optimization approaches [BAC09; BC10; SMFL00; ZC91] the surface albedo of an object can be estimated and used for illumination reconstruction.

2.7 Image and Video Augmentation

Augmentation of image and video data is the process of compositing additional information into the original data. One application of image augmentation is to add detail in image regions that have been deleted or need to be replaced or inpainted [BSGF10; LMG12] or lack desired resolution or level of detail [EESM10; EM10]. Alternatively augmentation may refer to embedding additional information, such as text overlays, context aware annotations, or overlaid information graphics, as for example in augmented reality applications for mobile devices like *Layar* [Lay] or *Google Glass* [Goo]. Altering the appearance of objects is also possible, e.g., by mapping virtual materials and textures to objects detected and tracked in the original scene [HE08]. In this manner it is possible to swap the material of the garments worn by an actor [HE09; SM06], yielding a realistically looking cloth material even with animated data. The important part of this kind of augmentation lies in the realistic rendering and composition of virtual data or objects into the original scene. Correct illumination and precise tracking removes cues a human would need to identify the virtual object. Without meeting these requirements, a realistic and believable embedding of virtual objects into a scene is not possible.

Interactions of actors and synthetic objects allow to further increase the credibility of the virtual objects. To realize these interactions, the actor has to be remodeled as a 3D model to be able to include him/her with the 3D geometry of the artificial objects. This is a common process in the movie industry, as a large amount of the modern movie content is computer generated imagery (CGI) and composed into the original movie footage including the actor's performance.

3 Monocular Pose Estimation

The first step towards a complete application for augmentation of humans in monocular video material is the identification and description of a human actor visible in given monocular video material. This description most importantly includes a realistic model of the person's 3D pose and motion.

3.1 Motivation

Given a monocular video without any additional information about the scene configuration, it is a hard task to reconstruct the human actor at a high level of detail to describe his/her pose as accurate as possible. Using color differences to the background or motion information [CCBK06; SCF09], an automated segmentation into foreground and background is possible. The segmented foreground area, being the silhouette of the person, can then be used to further derive pose and shape parameters of the person in the video. To make the automated image segmentation more robust, the person can wear a suit consisting only of select predefined colors. This way, an algorithm can be developed to quickly and robustly identify the important foreground regions. Using a set of a-priorily known features or markers, using the predefined color

palette, allows to quickly and robustly identify a person in the video and use additional information of the marker configuration to derive his or her pose parameters for an underlying skeletal model.

3.2 Overview

To infer a set of 3D pose angles for a given skeletal structure from only a single monocular input image, the person has to be segmented from the input image. Relevant information about pose and orientation is only located within the segmented foreground region defined by the actor's silhouette. The process of detecting these predefined features can, therefore, be constrained to the region of the silhouette. The markers found on the suit and their relative positions to each other have to be evaluated in form of a marker configuration descriptor. This marker configuration corresponds to a certain pose and is comparable to a pose parameter description. It can, therefore, be used to relate marker information directly to pose parameters of an underlying skeletal model. In addition to the configuration of relative spatial marker locations, the outline of the person's silhouette can also be integrated into the pose descriptor. This approach has a major advantage compared to other, silhouette-based techniques. Using additional marker information includes actual occlusion information into the reconstruction problem. Therefore, it is able to describe the pose more accurately, even in ambiguous cases [RNWM11]. For example, from a single silhouette it is not directly possible to infer the positions of the lower arms, if these are directly in front of or behind the torso. Also, one cannot distinguish the spatial order of these limbs and decide if one arm occludes the other or if

they are visible at all, Figure 3.1. The markered suit with its unique pattern introduces this information into a trained RVM pose regression model and allows for a stable differentiation of different poses having similar silhouettes.

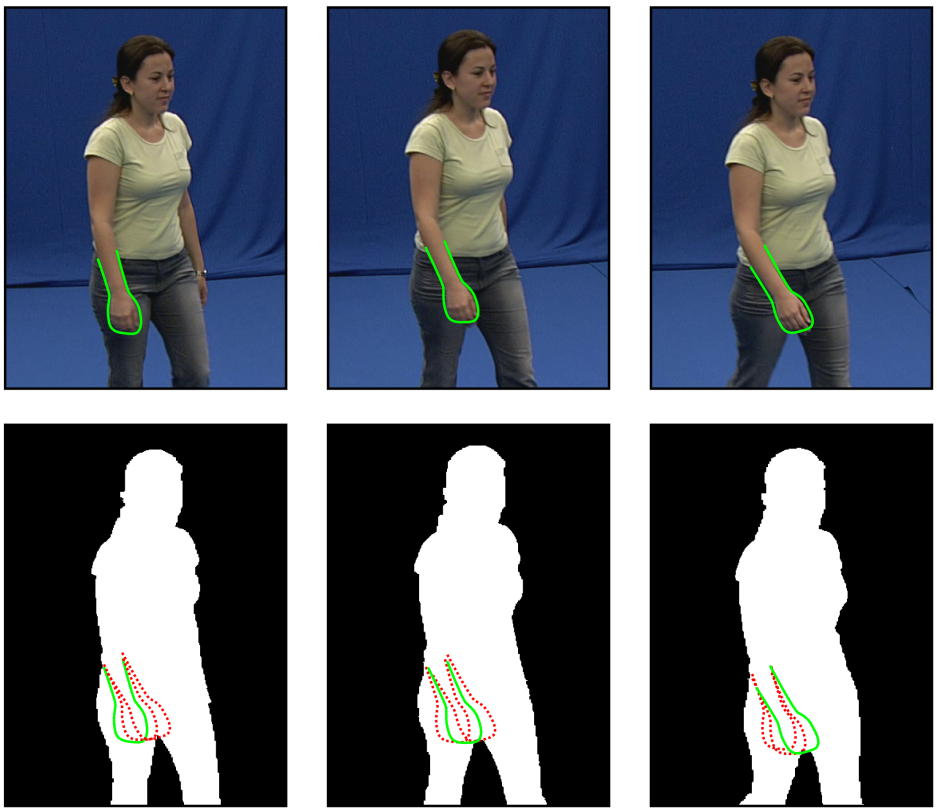

Figure 3.1: Silhouette Ambiguity: The images and silhouettes are taken from a regular walking motion. From the silhouette alone, it is not clear where the correct position of the arm is. The outline of the lower arm is shown in green for the corresponding silhouette and overlaid in red for the silhouettes of the other frames. Even though the silhouettes are very similar in this region, different arm position are possible. From a single silhouette, the actual position cannot be derived.

3.3 Silhouette Segmentation

To reduce computational complexity and allow for real-time pose estimation, the search space for potential marker locations used for describing the pose configuration needs to be narrowed down to the minimum region of interest (ROI). This region can be optimally described by the actor's silhouette, as it can be used to mask out all unnecessary background information. Additionally to the reduced search space, the silhouette can also give information about the current pose configuration in addition to the detected markers and their relative spatial positions. Therefore, as the first step of the processing pipeline an image-based silhouette segmentation seems logical.

The desired output of such a segmentation process is a pixel accurate binary mask for the actor in a given input image. For video data this mask should be temporally consistent, i.e. no flickering artifacts of the silhouette contour should occur. Adapting and modifying the live video segmentation technique of Criminisi *et al.* [CCBK06], a graph-cut based image segmentation can be implemented to optimally segment the marker pattern from any given background and also yield a stable and temporally consistent image segmentation for input video data. The segmentation technique uses likelihood look-up tables for various features per pixel in the image. For the chromacity, the neighborhood configuration, the local gradient, and the local motion vector of a pixel, likelihoods are estimated by using pre-computed lookup tables. Having a specially designed marker suit with predefined colors, these look-up tables can be easily learned from artificially generated sample data or even real and manually segmented recordings of an actor wearing the marker suit. A small set of about 20 consecutive sample

images with corresponding segmentation information was sufficient to train a stable segmentation model based on four separate likelihood priors represented by look-up tables.

These priors describe a pixel's likelihood to be foreground (FG) or background (BG) based on the following feature evaluations:

- C - The pixel's color represented in HSV color space
- N - The pixel's 3×3 neighborhood regarding chromacity, saturation and neighbor labeling
- G - The local horizontal and vertical gradient
- M - The segmentation label transition of a pixel over time regarding the local gradient
- T - The segmentation label transition of a pixel over time regarding a previous labeling

The color-based likelihood of a pixel to be labeled as foreground or background is learned directly from a sequence of manually segmented images,

$$C(\alpha, X) = -\sum_{n}^{N} log(p(c(x_n)|\alpha_n)) \tag{3.1}$$

where $c(x_n)$ is the color of a pixel in the n-th image X_n in the given image sequence, and $\alpha \in \{FG, BG\}$ is the segmentation label denoting foreground and background, respectively.

For the second prior term N the local neighborhood $N(x)$ of 3×3 pixels around a pixel x is evaluated regarding hue, saturation, and segmentation label difference

$$N(\alpha_x, \alpha_y, X) = -\sum_{n}^{N} \sum_{i}^{||N(x_n)||} log(p(d(x_n, y_{n,i})|\alpha_n, \alpha_i)). \tag{3.2}$$

Here, $y_{n,i} \in N(x_n)$ is a neighboring pixel to x_n, and $d(x_n, y_{n,i}) = (d_H(x, y), d_S(x, y))^T$ denotes the distance vector in hue and saturation of two pixels. This term allows to determine the likelihood of neighboring pixel labellings regarding a local neighborhood configuration, which is learned from the input training data.

As a third prior term the local horizontal and vertical gradient magnitude is evaluated. Based on these gradient magnitudes, the likelihood of a certain segmentation label to a pixel can be described via

$$G(\alpha, X) = -\sum_{n}^{N} log(p((g_h(x_n), g_v(x_n))^T|\alpha_n)) \tag{3.3}$$

where $g_h(x)$ and $g_v(x)$ denote a pixel's gradient magnitude in horizontal and vertical direction, respectively.

The fourth prior M is used to describe the likelihood of a segmentation label change over time regarding a pixels gradient magnitude $g(x)$

$$M(\alpha^t, \alpha^{t-1}, X) = -\sum_{n}^{N} log(p(g(x_n)|\alpha_n^t, \alpha_n^{t-1})). \tag{3.4}$$

This prior is especially important to control the temporal label consistency of the segmented regions, while boundary regions are likely to change the

segmentation label while other regions should be prevented from doing so during the segmentation computation.

The last prior is learned from the temporal label transitions $FG \rightarrow FG$, $FG \rightarrow BG$, $BG \rightarrow FG$, and $BG \rightarrow BG$ of a pixel x directly. The likelihood of a pixel to have a certain segmentation label can be determined by using the last two labellings of that pixel in a second-order Markov chain

$$T(\alpha^t, \alpha^{t-1}, \alpha^{t-2}) = -\sum_{n}^{N} log(p(\alpha_n^t | \alpha_n^{t-1}, \alpha_n^{t-2})). \tag{3.5}$$

These five priors can be learned using training data and then used for fast evaluation of a pixel's segmentation labeling. The likelihood values of each prior can be used as energies to a minimization problem describing the optimal image segmentation

$$\begin{aligned} E(\alpha^t, \alpha^{t-1}, \alpha^{t-2}, x, y) = \sigma_C \cdot C(\alpha^t, x) \\ + \sigma_N \cdot N(\alpha^t, x, y) \\ + \sigma_G \cdot G(\alpha^t, x) \\ + \sigma_M \cdot M(\alpha^t, \alpha^{t-1}, x) \\ + \sigma_T \cdot T(\alpha^t, \alpha^{t-1}, \alpha^{t-2}). \end{aligned} \tag{3.6}$$

This energy can be minimized using a standard graph-cut algorithm [FH04; RKB04]. The term evaluating the neighborhood configuration $N(\alpha_x, \alpha_y, X)$ is used to define pair-wise edge weights of neighboring pixels as these pixels can be represented as graph nodes. The other prior terms contribute to the edge weights connected to the two possible segmentation label nodes $\alpha_F G = FG$ and

$\alpha_B G = BG$. The individual energy terms are weighted using σ_C, σ_N, σ_G, σ_M, and σ_T and can be computed very fast, as the priors are represented as look-up tables. Only the computation of the local image gradients is required, which can be easily solved using standard Sobel filter convolution [SF68].

Using a GPU-accelerated implementation of the graph-cut algorithm [VN08] allows for real-time image segmentation. This segmentation is a binary labeling of the image into foreground and background regions, Figure 3.2, and serves as input for the subsequent processing steps.

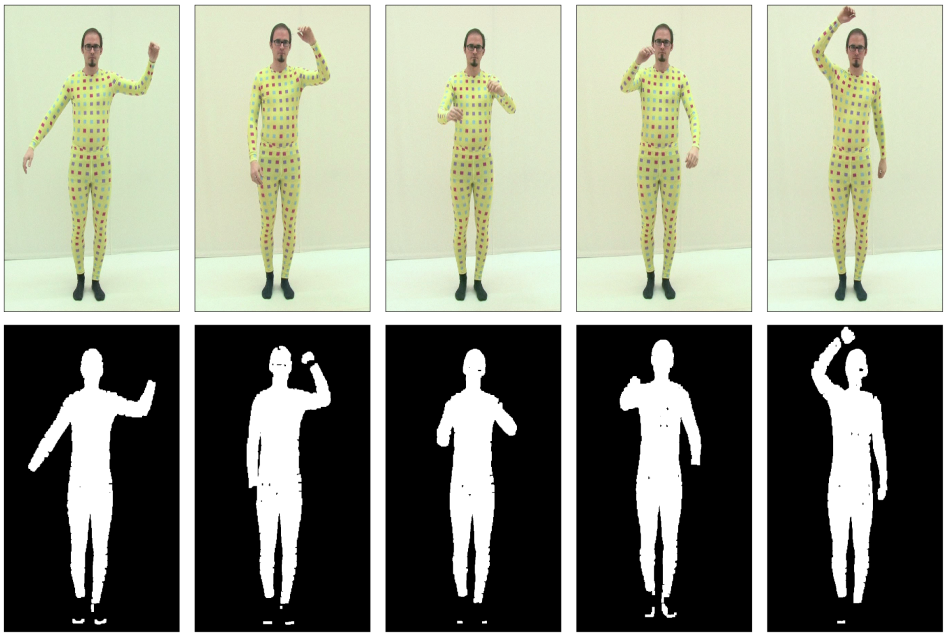

Figure 3.2: Color-based silhouette extraction: The proposed segmentation approach yields a robust mask describing the actor's silhouette. The mask is used in the subsequent steps of the pose estimation technique.

As the rest of the proposed pose estimation system is dependent on the used marker suit, it is no drawback to train the likelihood priors explicitly to the defined color palette of this special suit.

3.4 Pose Descriptor

The segmented actor silhouette can now be used to generate a special descriptor to describe the current pose of the actor.

The proposed pose descriptor is a vector of constant size. It is composed of information derived from the spatial configuration of detected markers, described in form of a *Feature Context* descriptor, Section 3.4.1, and information derived from the actor's general silhouette shape, represented by a *HOG* descriptor, Section 3.4.6

3.4.1 Feature Context Descriptor

The silhouette of a person provided by a manual or automated image segmentation already provides some important information of the actor's pose. The 2D position of the individual limbs describe the pose in general, while details about the exact position in space might still be missing. This is due to the missing information of the third spatial dimension. However, constraining the pose configurations to those being physically possible and probable, a reconstruction from silhouette information alone is possible, as Agarwal and Triggs have shown in [AT06] and their work on the *Shape Context* descriptor. This descriptor uses contour information in form of sample points and their local neighborhood

on the silhouette boundary to describe a human pose. Using a data driven approach based on actual human pose data, each descriptor can be related to a most probable pose of the skeletal model, see Section 2.3. A limitation of this technique is the inability to distinguish different poses resulting in similar silhouettes, see Figure 3.1. Only from a single silhouette, in certain cases, it is impossible to recover the exact location and orientation of body parts. All information of smaller body parts located inside the silhouette of a bigger body part are lost due to the nature of a silhouette projection. It is, therefore, impossible to infer whether e.g. a lower arm or a hand is located in front of or behind the torso. Also, the orientation of a limb may be ambiguous, as the exact pose in front of the torso is unknown in a single silhouette.

To overcome the limitations of the *Shape Context* descriptor proposed by Agarwal *et al.* [AT06], resulting from silhouette ambiguities, additional features on the body's surface have to be integrated into the descriptor. This allows to distinguish between different body parts and infer information about their local orientation and visibility. However, to properly identify these features with certain body parts they will need to be unique across the the used set of features.

Similar to the approach proposed by Scholz *et al.* [SM06] or the technique used in Microsoft's *Kinect* sensor, markers can by uniquely identified by using information about the markers in their local vicinity. Instead of using fully unique markers, a small set of N different markers can be used while the individual markers are uniquely described by the marker configuration in their neighborhood. While generating a pattern from these N markers, it must be ensured that the neighborhood configuration of a marker does not repeat itself.

This local neighborhood arrangement can be seen as an encoding scheme to the individual and unique markers.

3.4.2 Pattern Generation

As a body can move arbitrarily, the unique marker codes need to be invariant under rotation and, due to possible shear transformations of the markers placed on a suit, limited shear invariance is preferable.

To generate a valid marker configuration map, that can be used for a suit, a simple brute force approach can be used, as the process has to be executed only once for each used marker pattern and, therefore, is not required to perform at highly optimized computation times. Starting with the definition of the desired marker classes, i.e. the number of visually different markers, and the maximum desired pattern dimensions of width and height, a rectangular pattern is iteratively populated with random markers in a scan-line fashion. For every new marker candidate a check against all existing markers is executed, ensuring the new marker encoding is still unique. In case this check fails, another marker candidate is chosen. If no configuration of new markers generates a valid unique pattern, the process is started all over again using another random initialization and repeated until the pattern is completely populated with unique marker configurations.

The generated markers need to be optimally detectable in various camera setups and illumination conditions. They need to be colored and shaped in a way that ensures a reliable detection even under ill-posed conditions. Thus, too small markers would not be robustly detectable if the distance between

camera and actor is too large and the marker area covers only a small region of the image. Gaps between individual markers must be large enough to detect separate markers even in slightly blurred images, as this artifact may result from fast motions of out of focus captured images. Also, too big markers would affect the robustness of the pose descriptor relying on relative marker information. The bigger the markers are, the less markers can be used in a pattern of the same size and the less markers would add important pose information to the descriptor. As a consequence, a trade-off between the number and size of markers has to be found. Using an Full HD camera (1920×1080px) with a focal length of $35mm$ at an average distance of $2m$ from the actor, and assuming every marker to cover roughly 5-10 pixels in diameter, results in an optimal marker size of 2.5cm. As the complete pattern used for a markered suit only requires to cover the front and back of an average human, about 20×20 marker positions are necessary. I found 3×3 marker neighborhoods to be sufficient to uniquely encode the individual markers in this kind of pattern, Figure 3.3.

Using the constraints for rotational and shear invariance I was able to generate several markered patterns with unique neighborhood configurations with only three different marker classes, Figure 3.4.

Fast and efficient pose reconstruction requires a fast and robust feature or marker detection. Using shape coded markers like circles, triangles, or squares would make the pose reconstruction independent of the color of the input material. However, these shapes are rather complicated to robustly detect and distinguish under various rotations or possible partial occlusion. I therefore decided to use colored markers that can be properly identified using basic color classification algorithms. The color classification is not impaired by partial

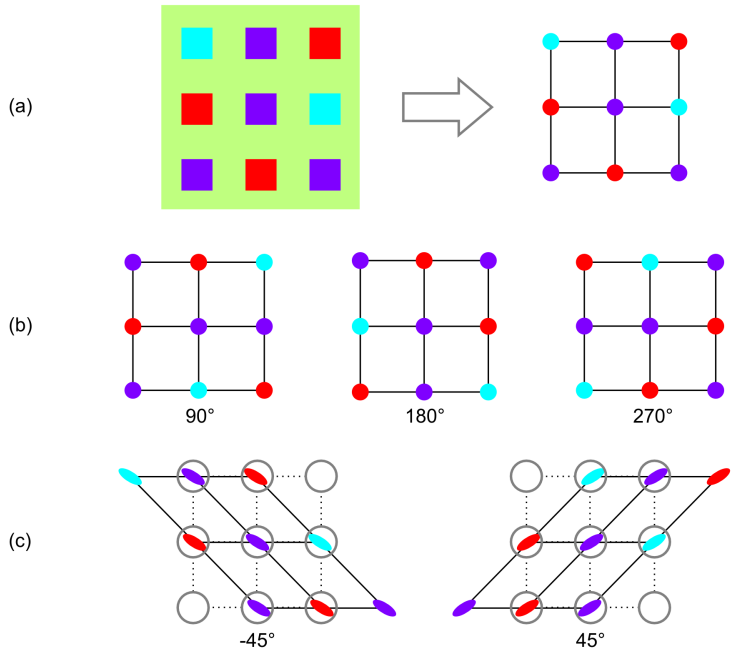

Figure 3.3: Unique marker configurations: The local 3×3 marker configuration (a) must be uniquely identifiable in various conditions and poses. Thus, the code must be unique even under rotations (b) or shear (c), as body parts may rotate and the suit may deform.

visibility of certain markers or their individual orientation. The marker colors can be classified by using colors that are optimally distinguishable. Using the HSV color space, colors can be differentiated by chromaticity (hue), saturation and lightness (value). As the body surface of the input video material may be affected by shadows or other shading artifacts, the lightness component cannot be used for a robust color classification, as this component directly describes the brightness of the color which is heavily influenced by external illumination and shadows. Therefore, colors are best distinguished by comparing their chromaticity or colorness. This color chromaticity may also be affected

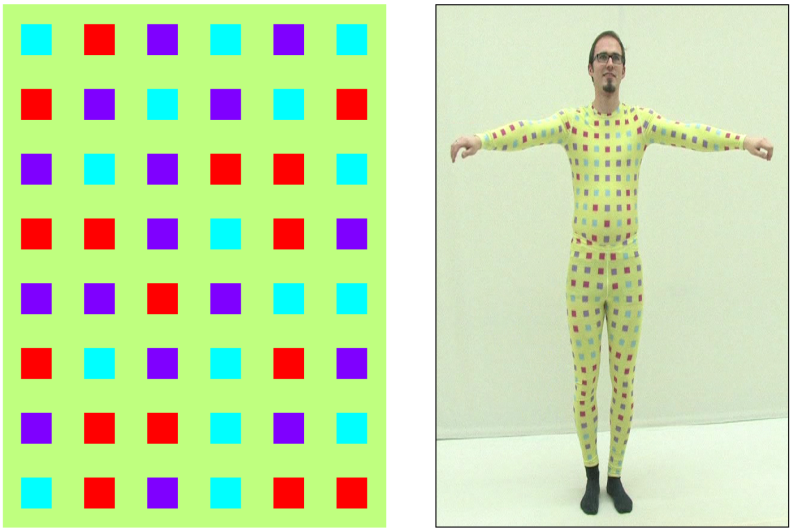

Figure 3.4: Full body suit using the unique marker pattern

by external light sources and their color, but can be corrected for using color calibration. Choosing marker colors that are equally distant in the polar hue-space of HSV colors provide an optimal basis for a robust color classification. Requiring only three different marker classes and an additional background color, only four colors with a hue-distance of 90° are necessary. To better differentiate markers from background, I decided to use a different saturation value for the background color. Additionally, I decided to use squared markers as the additional edge information, compared to circular markers, is beneficial for the final pose descriptor. While the unique marker encoding provides information about body part visibility and location, the marker edges add orientation information to the pose descriptor.

3.4.3 Marker Detection

To use the proposed marker patterns in combination with the marker based pose descriptors described above (Section 3.4.1) in a real-time application, the detection of the individual markers must be as computationally efficient as possible. As the markers in the generated patterns are designed to be optimally distinguishable by color, a fast method to classify a pixel's color has to be found. However, it is mandatory to only detect valid marker positions on the body's surface, not in the scene background. Therefore, to prevent many false positives for marker positions detected in the background of an image, the actor silhouette is used as the region of interest for the marker detection algorithm.

For a fast and robust marker detection, I chose to use individual image filtering steps for each marker color. Using per-pixel filters for each color class of the markers and the background, a likelihood map for every color can be generated.

Given a set of colors $\mathbf{C} = \{c_0, \ldots, c_i\}$ in HSV color space, representing the marker colors $c_0, \ldots, c_i$ as well as the suit's base color c_b, a per-color likelihood can be generated by simply computing a Gaussian likelihood $p(x_i, c)$ of each color $c \in \mathbf{C}$ for every pixel $x_i \in \mathbf{I}$ in the image

$$p(x, c) = e^{\frac{-d_H(H(x),\mu_H(c))^2}{2\sigma_H(c)^2}} \cdot e^{\frac{-(S(x)-\mu_S(c))^2}{2\sigma_S(c)^2}} \cdot e^{\frac{-(V(x)-\mu_V(c))^2}{2\sigma_V(c)^2}} \tag{3.7}$$

where $\mu_H(c)$, $\mu_S(c)$, and $\mu_V(c)$, and $\sigma_H(c)$, $\sigma_S(c)$, and $\sigma_V(c)$ are the mean values respectively standard deviations of the hue, saturation, and value of a marker class color c, while $H(x)$, $S(x)$, and $V(x)$ denote the color components of the

pixel color x. As the hue of a color in HSV color space is a polar value, the shortest angular distance between two hues is denoted by $d_H(c_i, c_j)$.

This fast evaluation yields separate likelihood maps $P(\mathbf{I}, c)$ for every marker color class $c \in \mathbf{C}$. The difference of a marker's color likelihood map and the background likelihood map $P(\mathbf{I}, c_j) - P(\mathbf{I}, c_b), c_j \in \{c_0, \ldots, c_i\}$ provides an even better classification likelihood for every marker class. These likelihood-difference maps are blurred by filtering them with a 2D Gaussian kernel. In the blurred maps the center pixels of the individual marker regions have a higher likelihood value than pixels closer to a marker's border. The center position for every marker is then computed by finding the local maximum in every marker region. This can be easily achieved using a per-pixel 2D filter, yielding a binary result denoting if the center pixel is the maximum value in the local filter window or not.

By fitting a 2D Gaussian to every marker region, a sub-pixel accurate marker center position can be computed, but this is not necessary as the marker positions are quantized as input to a discrete histogram-based pose descriptor, so sub-pixel accuracy is not required. The resulting pose descriptor would not improve by using more precise marker coordinates. Therefore, it is sufficient to use the simple filter-based technique to compute marker locations in image space in an easy and efficient manner. Eventually, for every detected marker region its computed center coordinate and color class are used for further processing.

3.4.4 Descriptor Generation

For every detected marker position, the relative position and distance to every other marker center within a predefined neighborhood region is determined. As the markers are placed on a suit, their visual size and relative distances will be proportional to the overall silhouette size of the person. Therefore, the region of interest (ROI) to select a marker's neighborhood is scaled according to the visual size of the actor's silhouette. This way, scale invariance of the pose descriptor is ensured. To reduce computation and descriptor complexity, the ROI should be small, while, on the other hand, as much information as possible about the marker neighborhood is preferred. A radius for a circular region of interest of $R = 15\%$ of the silhouette's height turns out to be sufficient and provides a robust pose description for all sample poses, Figure 3.5.

To describe the neighborhood configuration $N(m_i, R)$ of a marker m_i, a 3-dimensional histogram H_i is created using the relative distance, orientation and color difference as binning dimensions. This histogram uses a logarithmic scale to bin the distance dimension and is, therefore, represented by a log-polar histogram for every possible marker color, Figure 3.5. Using 5 radial bins for distance and 12 angular bins for orientation proved to be sufficient for robust pose description.

Every neighbor marker is inserted into the corresponding bin regarding its distance and relative orientation to the center marker, and its color class. To compensate for descriptor noise due to slight orientation changes of markers, soft binning is used to build up the histogram. This means that in case the marker properties do not exactly correspond to the center of the designated

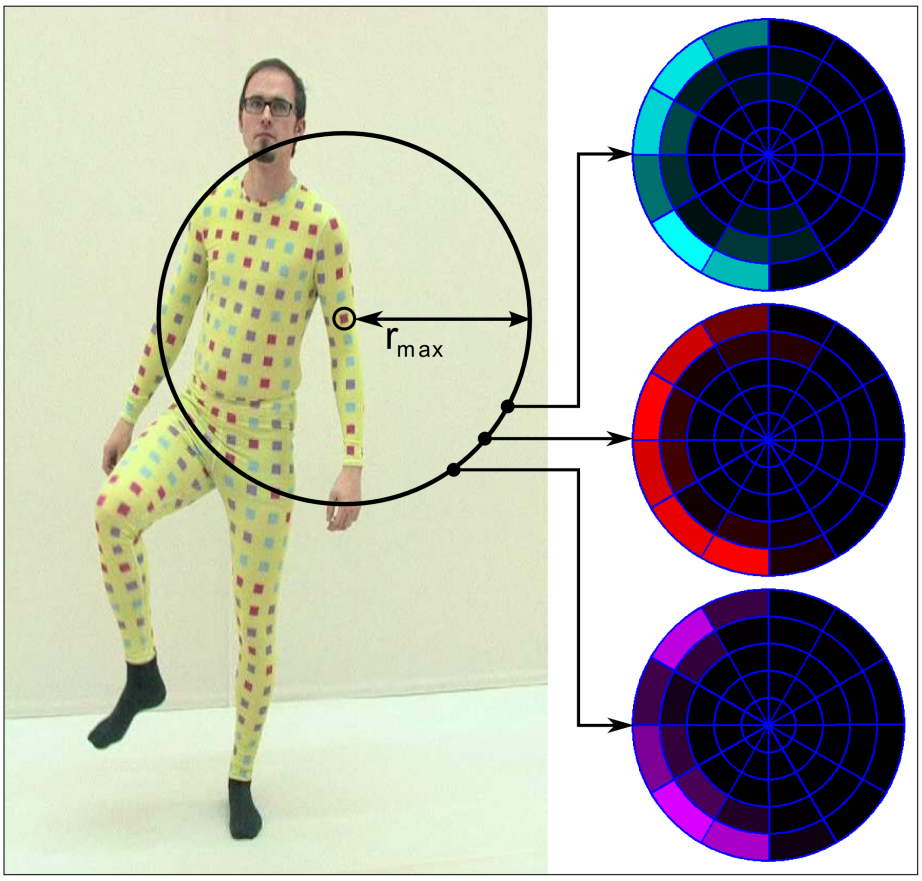

Figure 3.5: *Feature Context* Descriptor: The local neighborhood configuration of the detected marker positions and color classes is accumulated in a 3-dimensional log-polar histogram for every marker on the suit. The entirety of these histograms are combined in a common descriptor, describing the actor's pose as a whole.

bin the contribution of every marker to the histogram is spread proportionally across neighboring histogram bins. As a final step to building the per-marker descriptor histogram H_i, it is normalized to compensate for changing numbers of visible markers within the region of interest.

3.4.5 Descriptor Normalization & Compression

Even though the per-marker histograms have been normalized, a full pose description is only possible by properly combining the per-marker neighborhood information. As the number of generated marker descriptors is strongly dependent on the orientation and pose of the actor in a given frame, this information cannot be directly combined and used to describe and compare one pose to another. The size of the combined histogram data would change depending on marker visibility, while a feature vector for comparison purposes would need to be of constant size. Thus, the set of per-marker descriptor histograms has to be transformed into a uniformly sized pose descriptor.

Using a set of training poses and corresponding image data allows to precompute a large amount of possible per-marker neighborhood descriptors. I use k-means clustering [Mac+67] to identify the $N = 200$ most important clusters within the set of all possible per-marker descriptors. Having pre-computed these clusters, another one-dimensional histogram can be generated by comparing all per-marker descriptors of a given frame to these cluster centers. The distance to each cluster center can easily be described by the Euclidean distance of per-marker descriptor and cluster center as both vectors are of the same size. Using a soft binning strategy again, every per-marker descriptor votes for its 5 nearest neighbors in this one-dimensional histogram and, eventually, noise resulting from fluctuations of marker visibilities is reduced. As the number of per-marker descriptors H_i potentially varies per frame, this histogram is also normalized. The final result is a one-dimensional histogram of fixed size $N = 200$ for every possible input frame. In this form the pose descriptor H_{FC}

is a very condensed representation of all visible markers and their neighborhood configurations.

3.4.6 HOG Descriptor

The contour of the segmented silhouette as well as the structures on the actor's body surface give additional information about the orientation of the different body parts. As I chose to use square markers for pose-description using the *Feature Context* descriptor, these squares provide important information about their position and orientation. Using a *HOG* descriptor allows to describe the pose specific edge orientations with a 2-dimensional grid of fixed size, describing the most important edge orientations of every grid cell [DT05].

Using a standard edge detection algorithm [SF68] the orientation of all marker edges and the silhouette edge segments can be easily computed. This yields a pixel precise edge orientation map for a given input image. As background regions are not relevant they are explicitly omitted during the computation by masking them out using the actor's silhouette, Figure 3.6.

For normalization and to ensure scale invariance of the descriptor, the grid is scaled according to the silhouette's axis-aligned bounding box. A polar histogram is used to collect the per-pixel edge orientations for every grid cell. Using soft binning, the gradient magnitudes of the pixels are inserted into 36 angular bins corresponding to the orientation of the respective edge gradient. This yields a histogram of fixed size for every grid cell which describes the most important edge orientations in that cell.

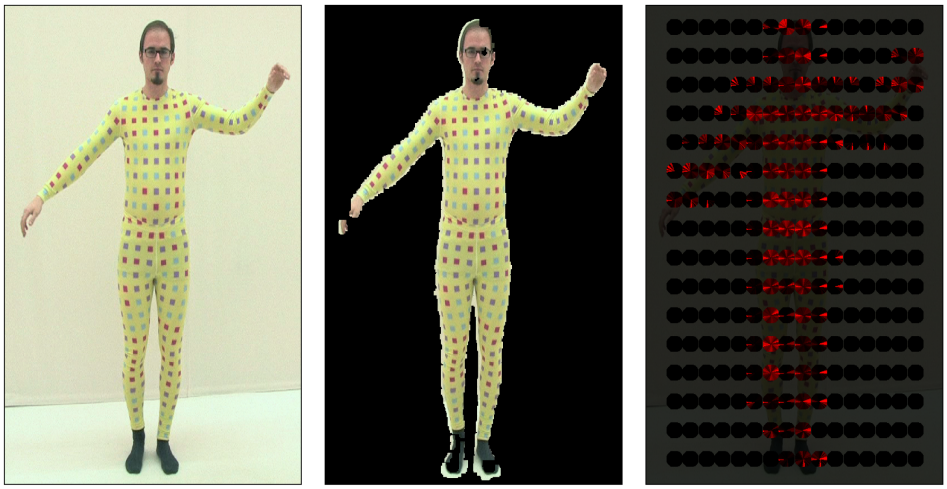

Figure 3.6: *HOG* Descriptor: The original frame (a) is multiplied with the segmentation mask of Section 3.3 to combine the silhouette gradients with the local marker gradients (b). Using an axis-aligned grid, scaled according to the actor's bounding box, the gradient histogram can be computed for every grid cell (c). The combination of all gradient histograms can be used as a HOG-based pose descriptor.

To further compress the *HOG* descriptor, a similar compression technique to the one used to generate the *Feature Context* pose descriptor is used. A fixed set of $M = 100$ reference gradient histograms is generated by k-means clustering all gradient histograms generated from training data. The histograms of each cell are now compared to the cluster centers, and a one-dimensional histogram H_{HOG} is built as final descriptor. This process discards information about the position of certain gradients orientations as the 2D cell index is removed from the descriptor, building a single histogram. However, the descriptor still yields good results and is able to describe different poses robustly.

3.4.7 Combined Pose Description

The combination of both descriptors, *Feature Context* and *HOG*, encode information about an actor's pose and the 3D orientation of body parts and allow to distinguish different poses robustly. In addition to the two feature vectors H_{FC} and H_{HOG}, the 2D location and size of the actor silhouette in screen space is integrated into the final pose descriptor vector x of size $U = N + M + 4 = 304$. For every image of an actor wearing the marker suit, such a pose-specific descriptor vector x can be generated by using only image-based filter operations.

3.5 3D Pose Regression

To derive the 3D pose parameters for a skeletal model from a pose descriptor as described in Section 3.4, a mapping of the descriptor space to the pose parameter space has to be found. Using machine learning techniques, this mapping can be derived from a training data set where a vector of 3D pose parameters is provided for each corresponding pose descriptor vector.

A Support Vector Machine (SVM)-based classifier to derive the desired values from a descriptor [Gun98] is viable only for single images but does not necessarily generate smooth motions for a sequence of frames. The result is the best matching vector of the training data set and its quality strongly depends on the number of samples used for creating the training data set.

To create stable and temporally consistent poses from image descriptors, Agarwal and Triggs [AT04; AT06] use Relevance Vector Machine (RVM) regression to map their silhouette based *Shape Context* descriptor to 3D pose data.

For every *Shape Context* descriptor, the best matching set of training poses is selected and an interpolated pose is generated.

To be able to generate smooth transitions between different sample training poses for a given pose descriptor, I also use *Relevance Vector Machine* regression, yielding spatially and temporally consistent poses that can be combined to obtain smooth and continuous motion.

3.5.1 Relevance Vector Machine Regression

Relevance vector machine regression is a way to derive a data vector from a given input feature vector. Instead of directly classifying this input vector, as in support vector classification, the relevance vector approach uses a multivariate classification and weighted interpolation of these classes for the final data vector and can be described by

$$y = X_f \cdot d(X_b, x), \tag{3.8}$$

where $x \in \mathbf{R}^M$ is a feature descriptor as input vector generated from the input data, and $y \in \mathbf{R}^N$ is the unknown data vector that needs to be reconstructed. The matrix $X_b \in \mathbf{R}^{M \times S}$ represents the set of S support vectors to pre-classify the input vector. For every support vector, the similarity measure d to the input vector is determined and used to map the input vector to the optimally fitting and interpolated data vector via $X_f \in \mathbf{R}^{N \times S}$. The resulting data vector is, therefore, a weighted sum of all support vector mappings, while the weights describe the similarity of the input vector to the individual support vectors. In order to yield optimal results, the RVM-based regression requires this pre-trained data model described by X_b and X_f to homogeneously cover all possible

pose configurations. This model can be explicitly learned from annotated training data consisting of pairs of the image based pose descriptors described in Section 3.4 and the corresponding angular configuration of a 3D skeleton pose. The required 3D skeleton pose data is provided by motion capture recordings. For a valid training data set, the motion capture data needs to be synchronized with the image data and the derived pose descriptors.

Courtesy of the *Drematrix* computer graphics lab at *HTW Dresden*, I was able to create a set of training sequences using a multi-view motion capture stage provided by *Organic Motion* [Org]. Using the marker suit (Section 3.4.2) and a full-HD camcorder, a video was captured in sync with the motion capture recording for every training sequence. The per-frame analysis of these video sequences yielded a pose descriptor for every frame, which was then assigned and directly mapped to the pose of the actor based on a skeletal model using 20 joints. It is described by a N-dimensional pose angle vector ($N = 60$) using three Euler-angles per joint. This labeled training data set is then used to train and evaluate a proper mapping function from the descriptor space to the desired pose parameter space.

As the training data generated from video recordings possibly contains many similar poses, I decided to apply a k-means clustering on the pose descriptors of all captured training poses. This yields the S most significant pose descriptors and their corresponding pose configurations of the underlying skeletal model. I chose to use four different cluster counts $S \in \{512, 768, 1024, 1500\}$ to find the optimal clustering for a proper RVM regression model training. For all S cluster centers of the available pose descriptors, the closest and, therefore, best matching pose descriptor was selected and inserted as a basis vector into the matrix X_b.

This matrix describes the basic support vectors for classification of a pose from pose descriptors. To make the pose reconstruction robust, I chose to substitute the original pose angle definition of a joint in polar space, where the numerical angular value gets wrapped around at the $2\pi = 0$ gap by an angle representation that is split up into sine and cosine part of the angle. Besides a continuous representation of joint rotation angles, the linear interpolation of two poses, being one of the major advantages of a RVM compared to SVM classification, is also more robust and correct using this angle representation. This joint angle representation is also suited better than a quaternion angle definition, as the poses described as quaternions require spherical linear interpolation *SLERP*. The used RVM model, however, does not support this kind of interpolation. In addition with the 3D location of the skeletal model this alternate joint angle description results in a pose vector y of size $V = 123$.

To finally recover a skeleton pose vector y from a given pose descriptor x using a pre-trained RVM model $\{X_b, X_f\}$, at first the Euclidean distance d of the pose descriptor x for all basis vectors in X_b is determined as a similarity measure. This yields a distance matrix $D = d(X_b, x) \in \mathbf{R}^{1\times M}$ containing the distances between the basis vectors in $X_b \in \mathbf{R}^{U\times M}$ and the pose descriptor $x \in \mathbf{R}^{U\times 1}$. This distance matrix D can then be used to derive the optimally fitting skeleton pose vector y by solving

$$y = X_f \cdot D^T, \tag{3.9}$$

where $X_f \in \mathbf{R}^{V \times M}$ provides a matrix of kernel functions. These describe the interpolation scheme of the poses related to the basis vectors in X_b regarding the weights described in D. This yields an optimally interpolated pose vector y.

In the application of the pose regression using the pose descriptor based on marker information, the pose descriptor vector $x\ in\mathbf{R}^{304\times1}$ is compared against the basis vector set in $X_b \in \mathbf{R}^{304\times S}$, yielding the distance matrix $D \in \mathbf{R}^{1\times S}$. This matrix describes the observed pose with respect to the S pose samples taken as basis vectors X_b for the RVM model. Using the matrix of kernel functions $X_f \in \mathbf{R}^{123\times S}$, the distance matrix D as a representation of the pose descriptor x can be mapped to a pose angle vector $y \in \mathbf{R}^{123\times1}$ describing the pose based on the skeletal model used for creating the training data.

As the matrices X_b and X_f are precomputed using the training data, the evaluation of a pose descriptor and the mapping to actual pose angles of the skeleton is just a series of matrix multiplications. These operations are fast and can be realized as a real-time pose reconstruction for a given pose descriptor vector.

3.6 Regression Results

For evaluation, I compared pre-captured motion capture data with the pose vector data generated by the learned RVM model. To make this comparison valid and to prevent the reconstruction of the actual basis vectors, I used a disjoint test sequence not used for training the RVM model.

For recording the initial training data, that was also partially used to train the silhouette segmentation model described in Section 3.3, a Canon XH A1

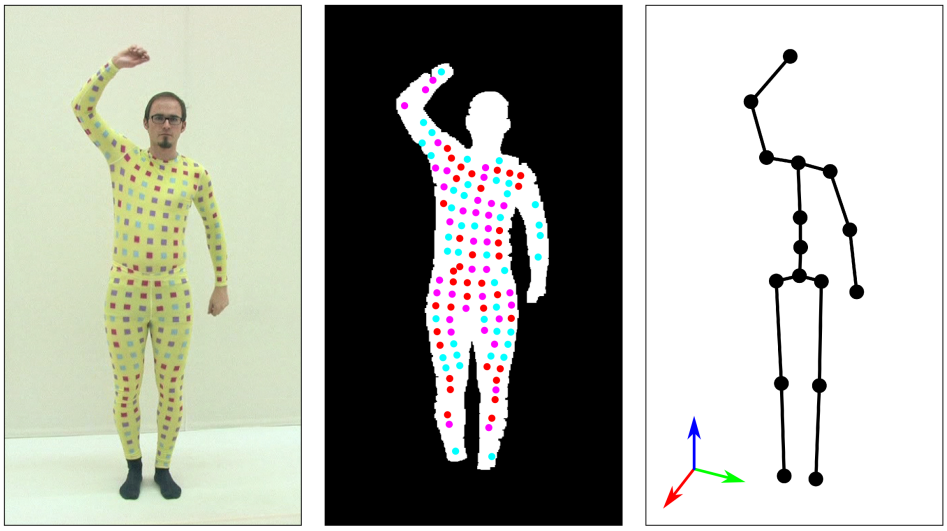

Figure 3.7: Pose regression result: From the original frame (a) all relevant data for the pose descriptor, i.e. the actor's silhouette and the individual marker positions, are extracted (b), and from the generated pose descriptor a 3D pose vector (c) is interpolated by the utilized RVM regression model.

Full-HD camcorder was used in combination with a 16-camera motion capture stage by *Organic Motion*. The reference video data of 5666 frames in total was recorded at a resolution of 1440 px an at a framerate of 25 fps.

For generating the silhouette segmentation model only 20 frame of this training data were sufficient, while a manual segmentation had to be provided. The RVM model for pose reconstruction was learned from a subset of the available 5666 frames, where the video capture was synchronized to the motion capture data using a reference frame and a subframe temporal synchronization technique similar to [MSMP08]. To every frame of this training data set, the silhouette segmentation technique was applied yielding a realistic silhouette that was then

used to generate the final pose descriptor vector. The set of pose descriptors could then be identified with the corresponding 3D skeleton pose angles provided by the motion capture system. As described in Section 3.5.1 the skeleton consists of 20 individual joints, each having three degrees of freedom (DOF). From these pairs, the most significant $S \in \{512, 768, 1024, 1500\}$ were selected for training the final RVM regression model in a Matlab implementation of Agarwal *et al.* [AT04].

For evaluation, a separate synchronized motion capture recording was used to create a *ground truth* data set that was not related to the basis vectors of the RVM regression model. This ensured that the process is not only able to detect a current input vector in the training data set, but to actually derive a plausible pose angle vector from any given pose descriptor.

The results can be seen in Figure 3.8, where the average angular error of the individual body joints is shown with respect to the number of samples used for training the regression model.

To evaluate the correlation between the size of the training data set and the final reconstruction quality, $S \in \{512, 768, 1024, 1500\}$ sample pairs of pose descriptors and pose angle vectors were used to train individual RVM regression models. As can be seen in the plot, Figure 3.8, the more samples used for training the model, the more accurate the pose reconstruction gets. However, this improvement is not very significant. For certain joints like the *neck*, *arms*, or *feet*, the angular error is significantly higher than for other body joints. As those joints are covered by only few markers, slightly different positions or rotations around the joints' major axis (around the bone) cannot be visually distinguished very well by the pose descriptor. The reconstruction improvement

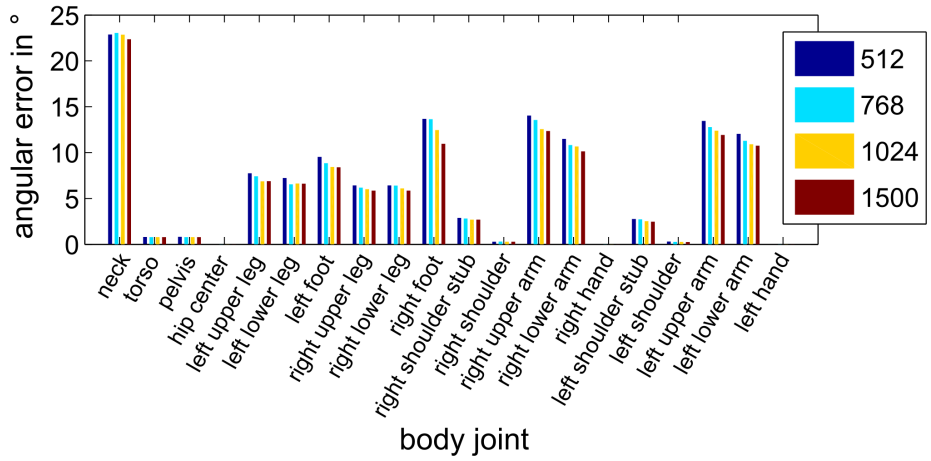

Figure 3.8: Average angular error compared to the ground truth comparison data set for every joint of the skeletal model. The quality of the pose reconstruction improves the more samples are used to train the RVM model. Still, the error is rather high for some of the body joints. These are joints that are not covered by many markers and do not vary much visually when rotations around the joint's major axis occur (e.g. *arms*, *feet* or *neck*).

for those joints, however, is still significant compared to the silhouette based pose reconstruction of Agarwal *et al.* [AT06]. As can be seen in the angular accuracy comparison Figure 3.9 between their approach and the proposed pose reconstruction technique, the angular reconstruction quality is better for all body joints, especially for the arms and legs. As these might overlap with the torso silhouette or each other, the marker-based pose description is superior to a purely silhouette-based pose reconstruction as it explicitly allows to distinguish between poses having ambiguous silhouettes.

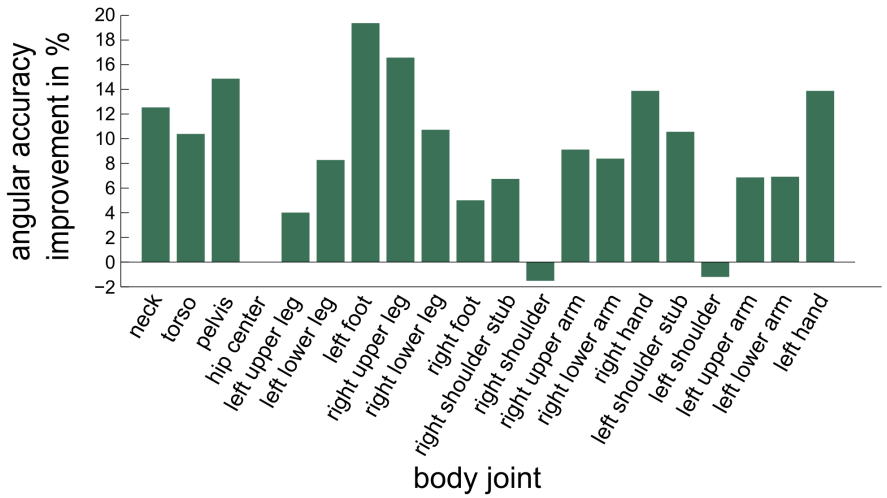

Figure 3.9: Improvement of the per-joint angular error compared to the silhouette-based technique by Agarwal *et al.* [AT06] when applied to the exact same data set. My proposed marker-based method improves overall reconstruction quality, especially the accuracy of the limbs. The reconstruction quality of the *shoulder* joints is slightly worse than that of the compared method, but these joints do not contribute much to the final visual pose.

Thus, even though the angular error of some joints was rather high, the overall quality of the reconstruction is still much better in comparison to existing techniques.

3.7 Acceleration of Pose Descriptor Generation

As the proposed technique is mostly based on image filtering operations and evaluations using pre-trained data models, the approach is optimally suited to be parallelized using multi-core computing hardware. The pixel-independent filter-

ing operations required for image segmentation and robust marker detection can be significantly sped up using massively parallel computing hardware [ZCW10]. The graph-cut processing step for the final silhouette segmentation, using the filtered image data as input, can also be solved using such an acceleration technique and multi-threaded computation. As Vineet *et al.*demonstrated with their *CUDA Cuts* implementation [VN08], solving complex energy minimization problems using graph-cuts is possible in real-time using modern day GPU multiprocessors. Furthermore, for demonstration purposes of complex image processing pipelines on adaptive and field programmable hardware, the segmentation and marker detection can be implemented as an adaptive computing processor on an FPGA [TE12]. The final pose reconstruction via RVM regression requires a precomputed RVM model and is evaluated using only simple matrix multiplications. Therefore, this step of the pose reconstruction is already very fast using a standard CPU.

3.8 Discussion

The proposed pose reconstruction technique is able to reconstruct 3D pose parameters from monocular video data. It is possible to generate such a pose parameter vector for a single image independently, but using a temporal filtering approach a temporally consistent and smooth animation can be reconstructed for a given video sequence. The proposed technique is largely based on independent and basic image filtering operations that are parallelizable and, therefore, suited to be executed on parallel computing hardware such as multi-core processors, data parallel processor architectures or even specially designed processing

pipelines modeled by configured FPGAs. The result is a real-time capable pose estimation technique.

In contrast to existing approaches [AT04; AT06], the proposed technique is able to reconstruct the pose of a person more stable. The novel, marker-based pose descriptor *Feature Context*as an extension to the *Shape Context* descriptor [BMP00] incorporates important additional information about occlusion and location of body parts that can be used during Relevance Vector Machine-based pose regression. This overcomes the limitations of silhouette and gradient based pose descriptors, heavily relying on the outer contour information a person, occurring for poses with ambiguious silhouettes.

A possible application can be found in the interactive design and prototyping process of garments. Instead of designing a prototype, manufacturing it, and testing it on a live model, the garment prototype can be designed and modeled virtually as a 3D representation and tested using an animated 3D body proxy that is controlled by the motion reconstruction of a model wearing the marker suit in front of a camera. This removes the part of actually producing real prototypes of a garment in early design stages and helps to speed up the overall development process.

It is also possible to quickly generate new motion capture data, if necessary, without the need of a complicated multi-view motion capture setup.

The potential real-time capability of the approach by using parallel processing makes it feasible for virtual mirror applications where people use the marker suit to animate a digital avatar in a virtual mirror-like environment. As an experimental application, the estimated motion was used to animate a virtual garment using a rigged body proxy, Figure 3.10. The proposed technique

could, therefore, be used in a virtual try-on system for an augmented shopping experience, where customers could virtually try on different clothes using a digital representation of themselves [DTE+04].

While the presented technique is able to reconstruct a pose from a single image, it requires the person in the photo or video to wear the marker suit that was initially used to train the RVM regression model. Therefore, it is still an *active* pose estimation technique, not capable of processing arbitrary and already existing monocular video data. A controlled capture setup is still required as the suit is a mandatory accessory to the motion reconstruction system. The suit, however, is rather cheap, as it is made out of a standard flexible fabric that was tailored and printed with the generated pattern by *Adidas*. Also, as illumination affects the recorded suit color, the global illumination during the video capture must roughly match the illumination used for creating the training data. With a strongly different or colored global illumination, the image segmentation and marker detection will fail, as they heavily rely on color information.

Having a controlled capture setup with correct illumination and the marker suit, a robust pose reconstruction is possible from monocular data. The overall reconstruction quality is superior to existing techniques. However, the proposed approach requires a marker suit and a RVM regression model based on training data, making it infeasible for arbitrary, existing monocular video material.

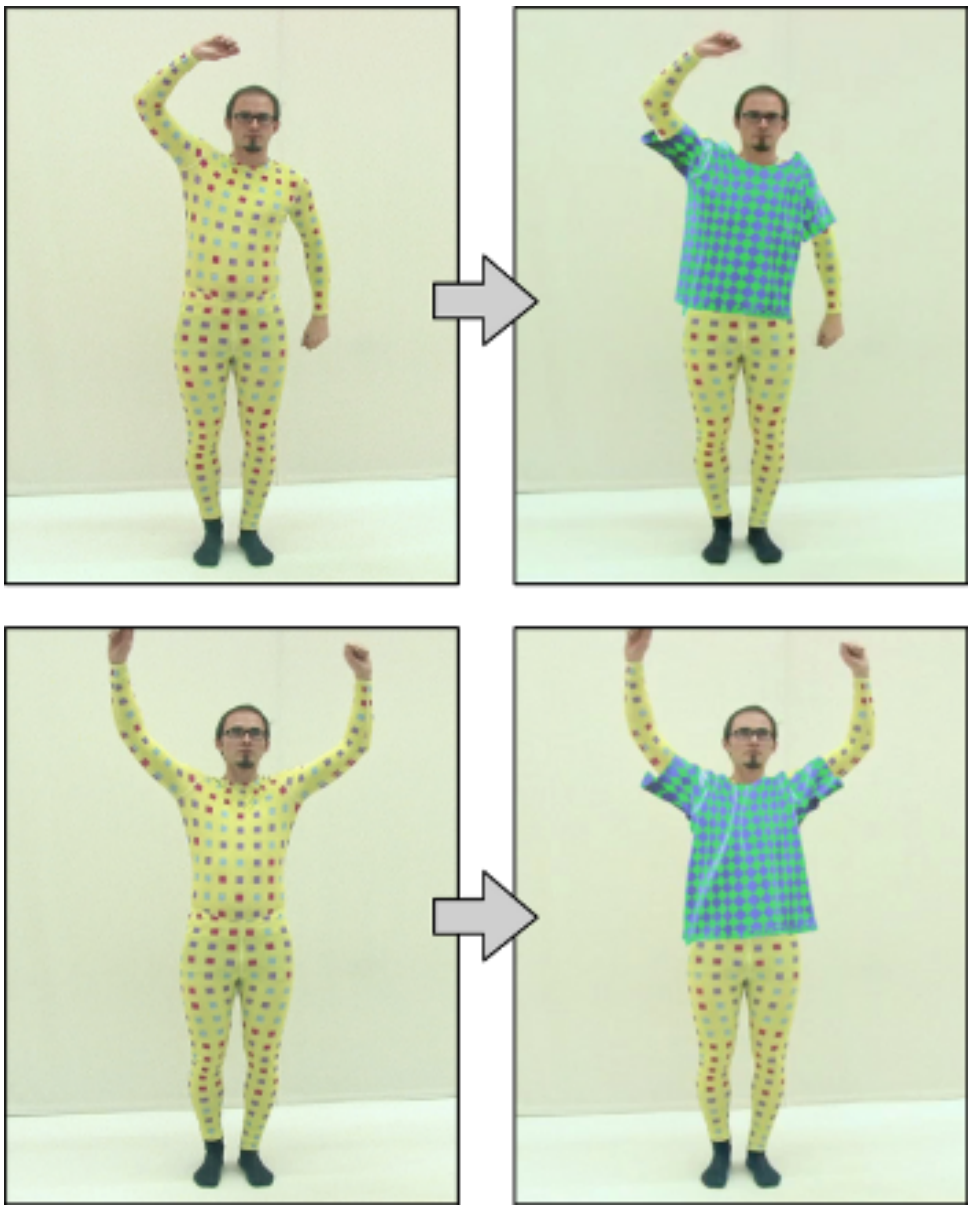

Figure 3.10: Exemplar application: The reconstructed motion can be used to animate a virtual garment. This way, a virtual try-on application can be realized.

4 Joint Estimation of Body Shape and Pose

To overcome the limitations of the active motion reconstruction approach presented in Chapter 3, another way to estimate an actor's pose from monocular input data has to be found. It should be possible to create plausible pose data without the need of a special marker pattern on a tight-fitting suit as a constraint to the video capture. Not having to rely on a special capture setup also removes the constrained availability of training data and having to pre-train a regression model to such a suit. This, on the one hand, requires a different pose and motion reconstruction approach. It would enable to process and augment already existing video data. Analyzing and processing existing video data removes the need for a real-time motion reconstruction application and allows to use computationally more complex reconstruction approaches. The offline processing aspect also allows to move the focus towards more *realistic* video augmentation, as the video analysis now allows to build up and reconstruct a more sophisticated model of the actor. Especially for realistic video augmentation, the model needs to incorporate a realistic estimate of the actor's body shape in addition to its motion. The desired result is a fully animated 3D body model combining a

smooth and detailed motion model with a realistic body shape representation that closely resembles the person in the original input video.

4.1 Silhouette-Based Reconstruction

To achieve the goals mentioned above, only information may be used for reconstruction that is available in any given video of a human actor. As textures of clothes and the body surface may vary, the most reliable source of pose and shape information is the actor's silhouette. As shown by Mori *et al.* [MM02] it is possible to properly describe a human pose from a silhouette alone. Based on a visual comparison of the original silhouette with a parameterized body model, the shape of the actor can matched by tuning the model shape parameters. Given the correct pose, this will yield the best fitting body shape to recreate the silhouette. In case of multi-view video footage, this approach allows to estimate a realistic body shape [CTMS03; HRT+09].

A body's shape can be approximately described using a parametrized body model such as *SCAPE* [ASK+05] or *MakeHuman* [Mak], giving control of the body shape via a set of deformation parameters $\Lambda = (\lambda_1, \ldots, \lambda_M)$. While the *SCAPE* model allows to control major body properties and shape parameters which are derived from statistical analysis of real world data, the *MakeHuman* body model additionally allows to control the shape and proportions even of small body details.

The silhouette used for approximating the optimal body shape can be created manually or by semi-automatically extracting it from the video data in a preprocessing step using a graph-cut based segmentation technique [RKB04].

The pose or motion, respectively, can be described using a hierarchical skeletal structure in combination with a blended vertex skinning of the 3D body model to this skeleton. Depending on the complexity of the skeletal structure, poses can be described more or less accurately by a set of joint angles $\Phi = (\phi_1, \ldots, \phi_N)$. The *MakeHuman* body model uses a skeleton composed of 31 joints, each having three degrees of freedom (DOF). The skeleton of the *SCAPE* body model uses a slightly smaller number of 22 joints which are limited to only a single DOF per joint. This makes the *MakeHuman* model more flexible as it allows to control the pose in more detail. The set of joint angles allows to fully describe a single pose and deform the skinned 3d mesh model accordingly.

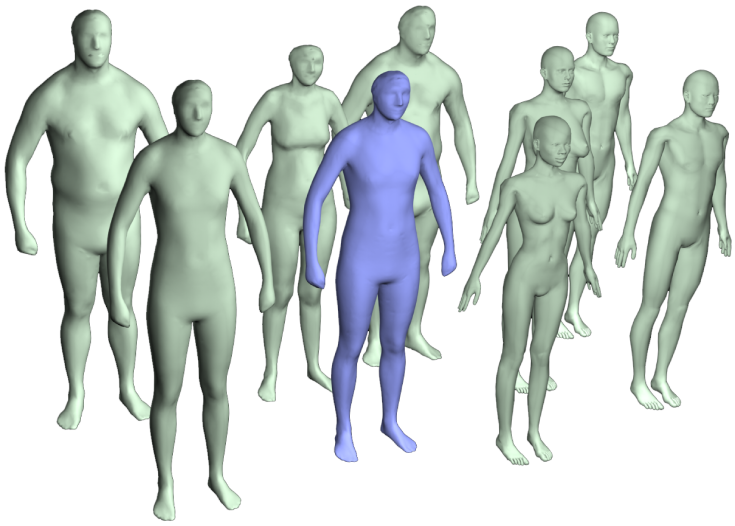

Figure 4.1: Parameterized Body Models: Exemplar renderings of both parameterized body models *SCAPE* (left) and *MakeHuman* (right) used for reconstructing the shape and pose of an actor. Both body models allow recreating a variety of body shapes.

As both parameter sets control the surface deformation of the 3D body model, the silhouette comparison allows to describe the matching quality of the body model to the real actor. The individual parameters can be optimized to reduce the silhouette matching error. Via energy minimization a plausible solution can be generated by finding a minimum of a fitting functional over the space of pose and shape parameters. This solution, however, might not be the correct solution, as ambiguities from missing 3D information introduce multiple local minima of the fitting functional. As both sets of pose and shape parameters are rather big, consisting of roughly one hundred single parameters, a direct fitting is computationally expensive and multiple local minima might exist that do not represent a valid solution. The body shape also does not change over time.To make the fitting procedure more efficient, I decided to use a disjoint optimization of either parameter sets. By fitting the body shape parameters first, these can be kept fixed during the body pose optimization. This reduces the dimensionality of the parameter space of the fitting functionals and allows to find an adequate solution more quickly.

4.2 Body Shape Estimation

The first step towards a properly reconstructed 3D model is the reconstruction of a realistic body shape. It is important to reconstruct the shape first as the reconstruction of the body pose heavily relies on a correct body shape representation.

Using an energy minimization optimization based on a simplex decent approach [NM65], a set of adequate fitting shape parameters can be estimated

for a given silhouette. The minimized energy is an error measure based on silhouette comparison between original and estimated body silhouette. The shape estimation problem can be formulated as an optimization of a set of shape parameters Λ based on this silhouette error.

Estimating body shape parameters based on this silhouette comparison, however, requires the pose in the input image to be known. So an initial pose Φ_α has to be provided to initialize body shape estimation.

4.2.1 Body Shape Error

The estimation of the body shape can be posed as the global optimization of the set of body shape parameters Λ regarding a silhouette-based energy functional $E_{\text{Shape}}(\Lambda; \Phi_t, \mathbf{M}_t)$ for a given input frame t

$$\underset{\Lambda}{\operatorname{argmin}}\, E_{\text{Shape}}(\Lambda; \Phi_t, \mathbf{M}_t) = E_{\text{s}} + \alpha E_{\text{h}} \tag{4.1}$$

which consists of two separate energy terms E_{s} and E_{h}. The overall silhouette mismatch E_{s} is represented by the absolute number of non-matching silhouette pixels and allows to describe the quality of a shape parameter set Λ. This silhouette mismatch is computed by re-projecting the body model $\mathbf{B}(\Phi_t, \Lambda)$ using the current body shape parameters Λ and initial pose $\Phi_t = \Phi_\alpha$ into the image plane and comparing it per-pixel with the segmented input frame $\mathbf{M}_t$

$$E_{\text{s}}(\Phi_t, \Lambda, \mathbf{M}_t) = \sum_{p=1}^{\#\text{pixels}} |\mathbf{M}_t(p) - P(\mathbf{B}(\Phi_t, \Lambda))(p)|. \tag{4.2}$$

The initial pose needs to be defined manually for a reference frame t. As some shape parameters may strongly influence overall body height, in certain cases a local minimum might be falsely considered by shrinking or enlarging the whole silhouette to compensate for shape mismatches at parts of the body's extremities, see Figure 4.2. The change of the height parameter might reduce the error term, while the correct solution would consist of tuning multiple parameters controlling the shape of select limbs instead. Therefore, the additional error term E_{h} is introduced to describe the height mismatch of the silhouette and further constrain shape parameter optimization

$$E_{\mathrm{h}}(\Phi_t, \Lambda, \mathbf{M}_t) = |y_{\min,\mathbf{M}_t} - y_{\min,\mathbf{B}_t}| + |y_{\max,\mathbf{M}_t} - y_{\max,\mathbf{B}_t}|. \tag{4.3}$$

For a common image or video, it can be assumed that the person to be reconstructed is standing up with his/her feet being the lowest point of the silhouette. From this, the relative difference in silhouette height between $\mathbf{M}_t$ and $\mathbf{B}_t$ can be used to describe the height mismatch of the current shape parameter set, while $y_{\min,\mathbf{M}_t}, y_{\min,\mathbf{B}_t}$ and $y_{\max,\mathbf{M}_t}, y_{\max,\mathbf{B}_t}$ represent the minimum and maximum y-values of non-zero pixels in the silhouettes of actor $\mathbf{M}_t$ and body model $\mathbf{B}_t$.

The comparison of the shape parameter estimation with and without the additional term E_{h} is shown in Figure 4.2. The value of $\alpha = 30$ yielded good results for all experiments.

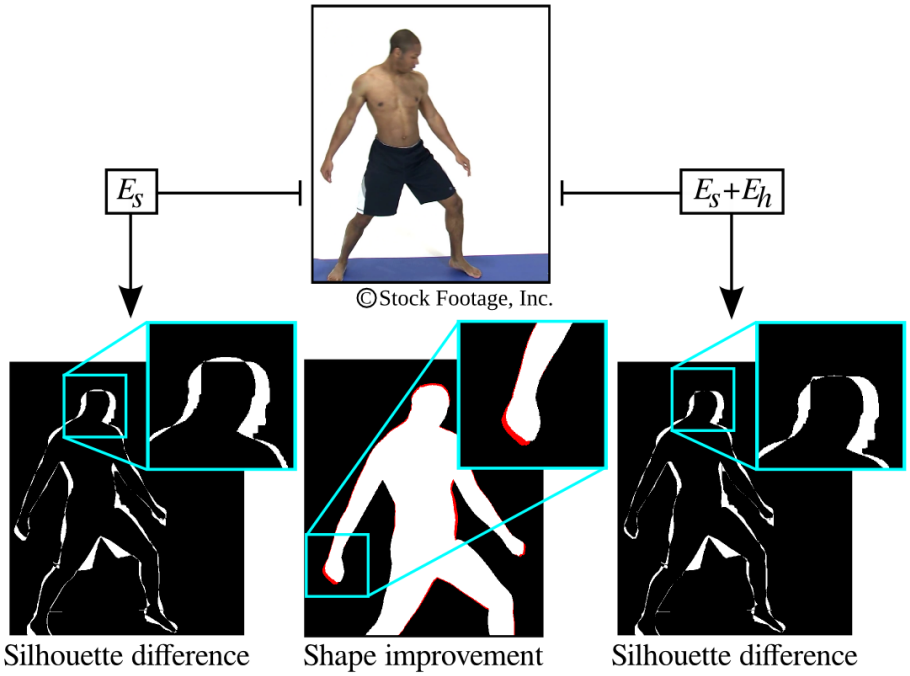

Figure 4.2: Silhouette Comparison: The silhouette overlap is evaluated using two separate quality measures. The term E_{s} describes the overall silhouette difference, while E_{h} penalizes differences in body height. The body shape optimization results with and without this additional term can be seen in the lower right and lower left, respectively. The body shape improvement is depicted in the lower central image.

4.2.2 Shape Parameter Optimization

The approach was tested with the *SCAPE* model as well as the *MakeHuman* body model parameterization. The former model provides a huge set of parameters describing the shape deformation in form of principal components of the underlying shape space data set. To reduce complexity I chose to use the $M_{SCAPE} = 30$ most significant shape parameters of this statistical model, as they already allow to control the body shape in detail, while the omitted parameters

are only of small influence. By optimizing all parameters in this subset, the described approach converges to a realistic body shape corresponding to the actor's shape in less than 500 iterations.

The *MakeHuman* body model was artificially designed by an artist and provides 50 shape parameters that directly relate to the shape of certain body parts. They are explicitly provided by the artist instead of being automatically generated by a PCA. Exploiting body symmetry, such as the shape of left and rights arms or legs, this set of body shape parameters can be effectively reduced to 35 unique shape parameters. Additionally, the *MakeHuman* model provides sets of so-called *MACRO* parameters. These parameters describe a semantic combination of body shape parameters, such as gender, overall size, and weight, and allow to speed up convergence to the best body shape. With a small set of $M_{MACRO} = 4$ of these *MACRO* parameters a quick convergence towards an initial, well fitting body shape is possible. The estimated values of the *MACRO* parameters can then be used as an initialization for a second shape optimization using more shape parameters called *MICRO* parameters, of which $M_{MICRO} = 50$ are used to tune the body shape in more detail. Using this two-staged optimization approach, in general, less iterations are necessary to converge to the most plausible body shape, even though the total number of parameters to be optimized for the *MakeHuman* model is larger than the number of those in the *SCAPE* model. The fast shape convergence to an initial solution using only 4 *MACRO* parameters is a big advantage of the *MakeHuman* model.

Compared to the *MakeHuman* model, the body shape description of *SCAPE* is directly derived from a principal component analysis of real-world body shape data. The first few shape parameters describe the shape deformation similar to

the *MACRO* parameters of the *MakeHuman* model, but there are no explicit parameters for single body parts as no semantic information was introduced into the model. Unfortunately, due to this unsupervised parameter space construction, certain shape parameters might also cancel each other out. A two-staged optimization is not practical as it is uncertain which shape parameters influence each other. Some semantic controls were added to the model by Hasler *et al.* by remapping the parameter space to a predefined semantic parameter space [HSS+09]. This, however, does not change the overall control of the body shape as the shape parameters always control multiple body shape properties in combination instead of single shape details related to distinct body parts.

Therefore, because the shape parameter mapping is more clearly and distinctive, the *MakeHuman* model provided better shape estimation results at faster convergence, even though the model is not based on range scan data of real people.

4.3 Body Pose Estimation

Body pose estimation makes use of an approach similar to the body shape estimation, ref. Section 4.2.

The set of pose parameters, defined by the degrees of freedom (DOF) of the underlying kinematic chain, is optimized via a simplex decent energy minimization approach. The energy functional uses also a silhouette-based energy term that describes the overall silhouette mismatch error. In addition to this silhouette error term, spatial and temporal information is included to ensure

that the reconstructed pose is logically possible and temporally consistent to poses of adjacent frames in the case of an input video.

4.3.1 Body Pose Error

The energy functional minimized during body pose estimation can be posed as follows:

$$\underset{\Phi_t}{\operatorname{argmin}}\, E_{\text{Pose}}(\Phi_t; \Lambda, \mathbf{M}_t) = E_{\text{s}} + \beta E_{\text{t}} + \gamma E_{\text{i}} \tag{4.4}$$

The first term is the same as in Equation 4.2 used for body shape optimization.

The second term E_{t} penalizes rapid temporal pose changes by evaluating the sum of absolute differences in angular acceleration of all body joints,

$$E_{\text{t}}(\Phi_{t-1}, \Phi_t) = \sum_{i=1}^{N} e^{\delta|\phi_{i,t} - \phi_{i,t-1}|} - 1, \tag{4.5}$$

where $\phi_{i,t}$ is the i^{th} pose parameter in frame t. Ambiguous poses can be eliminated using this temporal component as only poses are considered that are spatially close to a previous pose. Poses having similar silhouettes while being spatially different, e.g. mirrored along the viewing direction, are discarded early, preventing temporally inconsistent motion reconstruction. A weight of $\delta = 2.5$ for this error term yielded temporally consistent motions, while a larger value of δ falsely dampens fast motions of the actor.

As a third error term E_{i}, the geometric possibility of the pose is evaluated, and self-interpenetrations of the body model are penalized,

$$E_{\mathrm{i}}(\Phi_t, \Lambda) = \sum_{j=1}^{|\mathbf{V}|} d_{\mathrm{p}}(\mathbf{v}_j, \Phi_t, \Lambda). \tag{4.6}$$

For every body part, a body surface collision test is evaluated and for every vertex $\mathbf{v}_j \in \mathbf{B}_t$ the penetration depth $d_{\mathrm{p}}(\mathbf{v}_j, \Phi_t, \Lambda)$ is accumulated. This penetration depth is used to describe collision error magnitude. While $d_{\mathrm{p}}(\mathbf{v}_j, \Phi_t, \Lambda) = 0$ for all vertices $\mathbf{v}_j \in \mathbf{B}_t$ that are not penetrating any body surface, d_{p} is the perpendicular Euclidean distance to the closest surface point of the penetrated body part. Strong weighting of this energy term with $\gamma = 500$ ensures that self-interpenetrations are reduced to a minimum during energy minimization.

4.3.2 Pose Parameter Optimization

In contrast to the two-staged shape parameter optimization approach in Section 4.2, all pose parameters are optimized for together. The optimization is initialized with the manually defined initial pose $\Phi_t = \Phi_\alpha$ of the body model $\mathbf{B}_t$ and its estimated shape Λ from Section 4.2. Again a simplex descent based on the energy term $E_{\mathrm{Pose}}(\Phi_t; \Lambda, \mathbf{M}_t)$ described above is used to converge to a pose vector ϕ_t of the succeeding frame. This optimizer then proceeds to estimate the pose parameters Φ for the succeeding frames in sequential order. To optimize preceeding frames to the initial pose in frame $\mathbf{I}_t$, Φ_{t-1} can be substituted by Φ_{t+1} in Equation 4.5.

Due to the pose-dependent parts of the energy functional, the pose parameters can be weighted individually during optimization. This allows to explicitly dampen angular changes that might result in a large positional displacement, such as for body joints of the upper parts of the kinematic hierarchy. The

MakeHuman body model provides a very detailed kinematic model consisting of 31 individual joints, allowing to control individual fingers and multiple regions of the head, for example, while the *SCAPE* model only uses a coarse kinematic chain 22 joints to animate the model. Both models use three degrees of freedom (DOF) per joint. To reduce the parameter search space, unnecessary joints of the *MakeHuman* model are explicitly excluded from optimization. These are, for example, rotations along a bone's axis which would not be physically possible or do not affect the final visual hull of the body model. As these excluded degrees of freedom do not influence the final pose result very much and even might be undesired due to impossible poses, omitting these joints does not harm the optimization result. Reducing this parameter set to $N_{MakeHuman} = 53$ individual degrees of freedom, still being more than double the size of the $N_{SCAPE} = 22$ pose parameters of the *SCAPE* model, the optimization was able to converge to a plausible pose within $N \leq 200$ iterations for all test cases.

4.4 Initialization and Manual Interaction

The presented approach uses only silhouette information to recover a body's shape and pose from monocular input video data. The separate optimization of shape and pose requires the reconstruction algorithm to be initialized beforehand as no information about position and size is known a-priori.

As normally no detailed information about a person's shape is provided along with an input image, all necessary parameters controlling, e.g., gender, size, and muscularity, have to be derived from the image alone. For single images parameterized body models allows to create a realistic looking 3D human avatar

corresponding to the person in the image by providing a user-initialized pose and further optimizing pose and shape fully automatic [YLK11]. Using only silhouette information, it is possible to efficiently fit the body shape parameters $\Lambda = (\lambda_1, \ldots, \lambda_M)$ of the body model to match the appearance in the provided initial input frame.

From there, the estimated body shape parameters can be kept constant and can be used for pose reconstruction throughout the rest of the video sequence. Starting from the initial position Φ_0, the pose can be propagated to succeeding frames while optimizing the joint angles $\Phi_t = (\phi_{1,t}, \ldots, \phi_{N,t})$.

Due to orientation and ordering ambiguities of individual body parts in the resulting silhouette, the pose optimization might choose wrong local minima during the pose fit. To compensate for those fitting errors, I explicitly allow for the user to correct falsely reconstructed poses manually if necessary. Using a key frame-based animation model, it is easy to modify and correct pose parameters in these frames. This pose correction is then used as a guiding constraint in an additional pose optimization iteration. Linear interpolation between a user-defined key-frame and the last estimated pose configuration Φ_{t-1} influences the smoothing term and therefore guides the motion reconstruction according to the user constraint. These user-defined key-frames are possible per individual joint, minimizing the overall user interaction when correcting for a pose misalignment of a single joint or part of the kinematic chain.

4.5 Results

Given a monocular input video and a corresponding silhouette segmentation, it is possible to reconstruct a 3D body shape and motion model. After a manual pose initialization of the default body model, the body shape is estimated based on a silhouette-guided energy minimization technique. The estimated body shape is used to reconstruct the pose in all frames of the input video. Again, the parameters are optimized by minimizing an energy functional based on silhouette information, as well as temporal motion consistency and physical plausibility.

The reconstructed animated body model visually matches the shape and motion of the person in the original video and can be used for video augmentation purposes. Relying on silhouette data only, the technique can be applied to arbitrary video data. This way, already recorded footage can be modified in a post process and even videos from online video platforms such as *YouTube*™ or *Vimeo*™ can be processed, Figure 4.3 and Figure 4.4.

4.6 Discussion

The presented approach to shape and pose estimation is purely based on silhouette information. By comparing the original actor silhouette with the projected body model, shape and pose parameters of a parameterized body model can be optimized for in an iterated process.

In contrast to state-of-the-art techniques [JTST10], the proposed method does not rely on additional feature tracking ensuring temporal consistency as

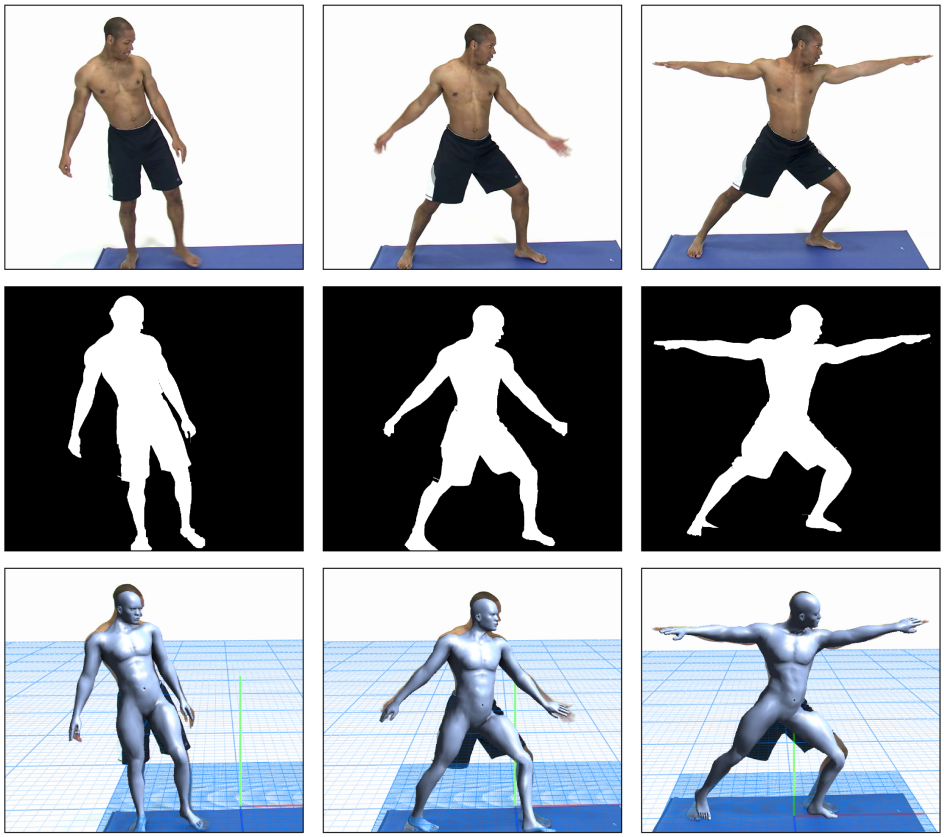

Figure 4.3: Shape and Pose Reconstruction Results: Using an online video provided by ©Stock Footage the shape and pose reconstruction can be demonstrated for arbitrary video data. The body model shape was properly estimated by the automated shape optimization of Section 4.2 to resemble an athletic male body, and properly adapted to the motion of the actor over a sequence of 299 frames using the semi-automatic pose estimation described in Section 4.3.

this is already included in the pose estimation error term. Also impossible poses are explicitly penalized by checking for body self-interpenetrations during optimization.

Figure 4.4: Shape and Pose Reconstruction Results: Using the *Into The Blue* ©MGM sequence, the 3D body of the female actress was modeled automatically.

Similar to [JTST10] the approach requires precomputed silhouette data using semi-automatic state-of-the-art segmentation techniques, and manual interaction for pose initialization and optional error correction. Therefore it is limited to offline processing and not real-time capable. However, for high quality video augmentation including realistic virtual garments, this is not necessary as the cloth simulation is another limiting factor requiring offline processing.

A high quality video augmentation, however, requires the reconstructed body model to perfectly match the original actor. As the parameterized body model does not describe additional clothing or hair, the estimated body shape may still mismatch the actor in certain body regions. Especially for wide and loose clothing the shape and pose optimization does not create acceptable results,

limiting the approach to video data with actors wearing rather tight fitting clothes. Slight shape mismatches, however, can be compensated for in subsequent image-based corrections.

5 Reconstructing Scene Illumination

A crucial component required for *realistic* video augmentation is a proper illumination model. This model is used to realistically illuminate and shade artificial objects embedded in the original video for augmentation. Such an illumination model, consisting of light source positions and light colors, has to be recovered from the visual appearance of the input sequence. These parameters are hard to estimate, especially when 3D information about the scene or reference objects are missing. Surface orientations are unknown in monocular video, making it impossible to relate surface shading directly to the direction of incident light. Also, surface material appearance is already a combination of surface material properties and illumination. Properly separating surface material properties and scene illumination from monocular data without any prior knowledge is a long standing problem [BC10; CWJ11; ZC91].

5.1 Motivation

Given a 3D human model reconstructed from the monocular input video, the additional information about surface orientation and location helps to overcome these limitations. Inspired by photometric stereo approaches and shape-from-shading techniques that recover surface normals from an image under known illumination conditions this principle can be inverted yielding illumination parameters from a given surface model, instead. However, without knowledge about actual scene illumination, the surface appearance cannot be decomposed in surface material and surface shading easily, as surface albedo is usually unknown. Therefore, a robust technique is needed to recover surface albedo from complex textured objects in monocular images or videos without any knowledge about the scene illumination. There are approaches trying to recover both, surface albedo and incident illumination, by solving a combined problem [GVWT13; VWB+12], but this increases the complexity of the overall problem. Instead, I propose to estimate the surface albedo using a statistical color analysis prior to the reconstruction of scene illumination from surface orientation and the estimated surface albedo.

5.2 Surface Albedo Reconstruction from Animated Geometry

The idea of this approach is to exploit 3D animated geometry corresponding to an object in an input video. Due to the object's motion it is likely that the illumination of most surface points changes over time. This is similar

to multiple illumination samples used in photometric stereo techniques and yields information about the influence of the scene illumination to the surface appearance. Sampling the colors for every surface point over time allows to analyze the change in visual appearance and eventually to separate illumination from surface albedo. A naïve approach would be to average the color samples of each surface point and use this as albedo estimate. However, this considers every surface point to be independent and does not properly model uniform surface materials that cover larger areas of the object. The result would be a very noisy surface albedo that might also be heavily influenced by local shadow artifacts. Also, invisible surface areas cannot be reconstructed due to missing color samples.

To remove the influence of local illumination artifacts and reduce noise, I propose an albedo reconstruction technique consisting of three steps.

- Color samples are collected for all visible vertices over the entire video sequence, Section 5.2.1.

- A two-staged clustering method is used to compute the most plausible surface colors, and all visible vertices are labeled according to their most probable surface color, Section 5.2.2.

- In a final step, the surface color of all invisible mesh vertices is recovered by choosing the best fitting surface color from their neighborhood, Section 5.2.3.

Using this proposed technique it is possible to recover the surface albedo of multi-material objects in monocular video sequences and to provide a complete HSV albedo map for the 3D mesh representation of this object.

5.2.1 Surface Color Sampling

Given an input video sequence $\mathbf{I}$ and a corresponding animated mesh, color samples for visible mesh vertices can be taken from every video frame $\mathbf{I}_t \in \mathbf{I}$. To reduce the complexity of the sampling I decided to use a per-vertex color sampling as this data is sufficient to reconstruct scene illumination. High frequency details of the surface material are not very important to solving this problem, as illumination affects a local surface patch uniformly. This allows to average the illumination influence of every local patch, which is equivalent to low-pass filtering surface material reflectance.

For every vertex $\mathbf{v} \in \mathbf{V}_t$ of the body model mesh, its visibility can be computed for every frame. This is done by projecting each vertex $\mathbf{v}$ into camera space using $\mathbf{v}' = \mathbf{P}_t \cdot \mathbf{M}_t \cdot \mathbf{v}$, with $\mathbf{M}_t$ and $\mathbf{P}_t$ being the modelview and the projection matrix of the frame t, respectively. The visibility for each vertex $\mathbf{v}$ can then be computed by comparing the projected vertex's depth $\mathbf{v}'_z$ against the depth map $\mathbf{D}_t$ of the rendered object and all color samples with $\mathbf{v}'_z > \mathbf{D}_t(\mathbf{v}'_x, \mathbf{v}'_y)$ are pruned.

To reduce the noise of the sampling data, color samples at vertices with a surface normal $\mathbf{n}$ almost perpendicular to the viewing direction $\mathbf{d}$ are omitted. These vertices belong to border regions of the projected object and, due to the visually plausible but not exact reconstruction of the body mesh, the projection

error of original and reconstructed surface point in image space might be too large to take valid color samples. I found a threshold value of $\alpha_{min} = 10°$ between surface normal $\mathbf{n}$ and viewing direction $\mathbf{d}$ to be sufficient to reduce the projection error-induced sampling noise.

Thus, only reliable color samples $\mathbf{c}_{\mathbf{v},t} = \mathbf{I}_t(\mathbf{v}'_x, \mathbf{v}'_y)$ are taken into account for frame $\mathbf{I}_t$. Finally, a set of color samples $\mathbf{c}_{\mathbf{v},t}$ is generated for every vertex $\mathbf{v}$ of the mesh surface.

In the following steps of the albedo reconstruction, the HSV color space is used in order to distinguish and classify color samples in chromaticity or hue and saturation. As there is no information about the scene's illumination, one needs to exploit the known geometry for every frame of the input video to compute a per-vertex ambient occlusion ratio [PG04], which describes the influence of the environmental illumination on each vertex. We normalize the brightness component V of all color samples $\mathbf{c}_{\mathbf{v},t}$ according to the precomputed ambient occlusion ratio $\mathbf{o}_{\mathbf{v},t}$ for each vertex $\mathbf{v}$ and frame $\mathbf{I}_t$.

5.2.2 Statistical Albedo Classification

To recover a surface material model from the collected color samples $\mathbf{c}_{\mathbf{v},t}$, I propose to use a two-staged clustering of the sample data that removes the influence of local illumination artifacts such as shadows or specularities while creating a set of material albedo candidates of predefined size.

All color samples are clustered regarding to their hue component first. I make use of k-means clustering [Mac+67] to segment all collected color samples into k_{H} groups with common hue values. Since the hue component in the HSV color

space is defined in polar space, $h(\mathbf{c}_{\mathbf{v},t}) \in [0 \dots \pi]$, projecting the one-dimensional polar hue sample values into 2D Cartesian coordinates, $h_x(\mathbf{c}_{\mathbf{v},t}) = \cos(h(\mathbf{c}_{\mathbf{v},t}))$, $h_y(\mathbf{c}_{\mathbf{v},t}) = \sin(h(\mathbf{c}_{\mathbf{v},t}))$ allows to properly compute the hue clusters in this 2D mapping. The 2D cluster center positions are then re-projected back into polar coordinates, yielding the k_H most prominent material hues visible in the input data. Every color sample is then assigned to one of these hue clusters.

To account for albedos with similar hue but strong differences in saturation (S) and/or value (V) (e.g., red, brown, and black), I make use of a subsequent k-means clustering to segment all color samples $\mathbf{c}_{\mathbf{v},t}$ associated with a hue cluster $\mathbf{C}_{\mathrm{H}_j}$ into k_SV sub-clusters regarding to their saturation and value components. These clusters describe the different shades of a material with a certain hue visible in the input data.

In combination $k_\mathrm{H} \cdot k_\mathrm{SV}$ HSV clusters $\mathbf{C}_{\mathrm{H}_j\mathrm{SV}_k}$ are generated of which the center coordinates are used as albedo candidates. The optimal number of clusters depends on the visual appearance of the object in $\mathbf{I}$, but I found $k_\mathrm{H} = 20$ and $k_\mathrm{SV} = 3$ to generally yield good results, as this describes a variety of possible hue values at different shading levels.

Having a HSV cluster labeling for all color samples $\mathbf{c}_{\mathbf{v},t}$, the associated cluster centers can be assigned as albedo color to every corresponding vertex $\mathbf{v}$. Since a vertex may have generated a variety of color samples with different associated albedo labels, for every vertex the cluster center albedo is chosen that was assigned to the majority of color samples of this vertex. This way the most likely albedo is assigned to each vertex and temporal, local changes to the visual surface appearance, like moving shadows, are suppressed.

5.2.3 Surface Albedo Reconstruction

The clustering in Section 5.2.2 generates and assigns the most likely albedo colors to the majority of visible vertices. However, for all invisible surface areas, such as object backsides or permanently occluded areas, and also for areas that did not provide reliable color samples no albedo color could be assigned. Assuming that the surface material is locally smooth, albedo colors for those surface areas can be derived from local neighbors with a valid albedo assignment.

An iterative hole-filling approach allows to quickly generate a complete albedo coloring of the animated mesh model. For every vertex $\mathbf{v}$ without an assigned albedo color, the albedo of the closest adjacent neighbor vertex $\mathbf{v}_j = \min_{\mathbf{v}_i} d(\mathbf{v}, \mathbf{v}_i), \mathbf{v}_i \in N(\mathbf{v})$ is selected, with $d(\mathbf{v}_a, \mathbf{v}_b)$ being the Euclidean distance of two vertices and $N(\mathbf{v})$ being the set of adjacent vertices to $\mathbf{v}$. The assignment is iterated in a region-growing fashion until all vertices have been assigned an albedo color.

The result is a complete vertex coloring of the input mesh model corresponding to the input video sequence. Note that due to the likelihood assignment of albedo colors to each vertex, not all generated albedo candidates are necessarily used in the final vertex coloring. Therefore, using a predefined, fixed number of cluster centers k_{H} and k_{SV} during albedo estimation does not heavily affect the final result. This would be the case, if the object exhibits more differently colored materials than there are allowed hue clusters k_{H}, but, in general, articulated objects only feature a small number of different materials.

5.2.4 Evaluation

To evaluate the resulting quality of the proposed technique, I used a synthetic test sequences for comparison of original and reconstructed surface albedo. This sequence features an animated human body model with predefined surface albedo that is rendered using a non-trivial scene illumination using several differently colored light sources, Figure 5.1.

After recovering the surface albedo from the synthetic sequence, a qualitative comparison shows that the estimated albedo is relatively constant over all input frames, Figure 5.2.

The accuracy of the estimated hue and saturation components is always above 95%. The relatively low quality of the estimated luminance component stems from the impossibility to decompose the brightness of the visual color appearance into environmental illumination and albedo brightness without any additional information. Assuming the illumination to be at maximum brightness, the darkening factor fully affects the albedo estimate instead of being a combination of illumination and albedo. Due to a low environmental illumination of the synthetic rendering the estimated color brightness of the albedo color is reconstructed as a scaled version of the ground-truth albedo luminance component, and the brightest estimated surface albedo is scaled according to the brightest visible color of the input images. Thus, the estimated image visually matches the original input in brightness. Due to the scaled albedo estimate the reconstruction error for dark colors is less than for bright colors, as can be seen in the visualized albedo estimation error for the first frame of the synthetic sequence, Figure 5.3(V).

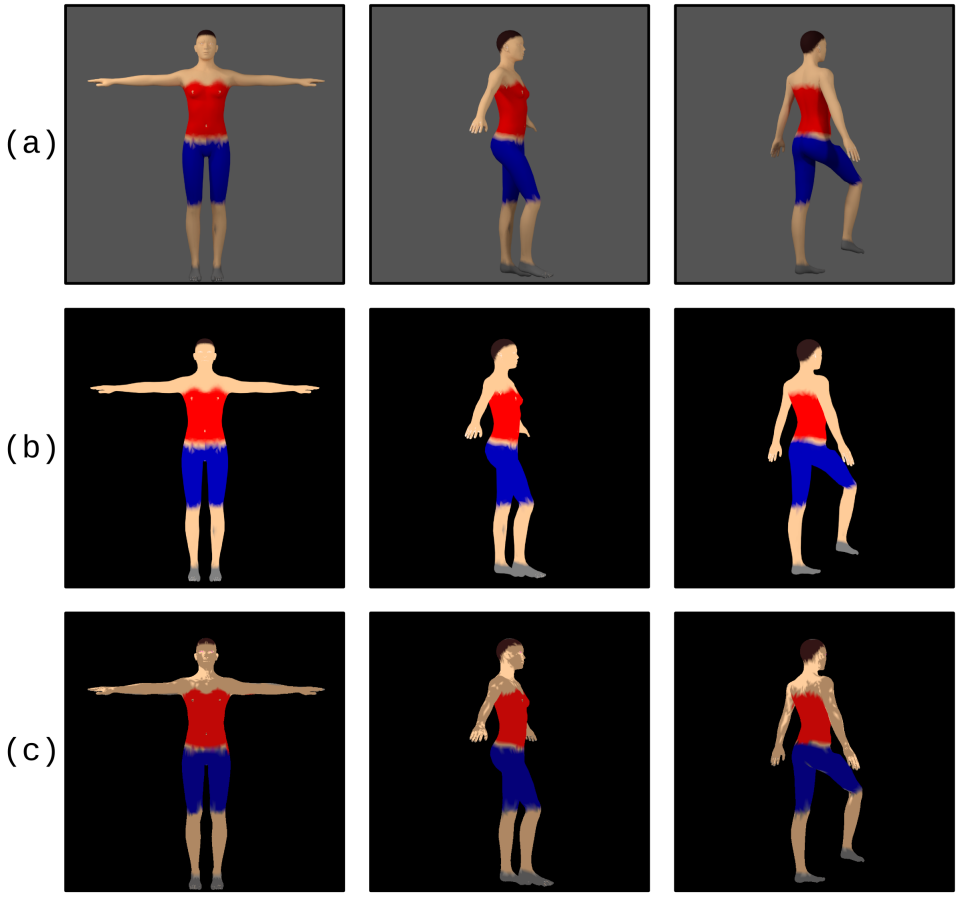

Figure 5.1: Example frames of the synthetic test sequence: The frames rendered with a path tracer can be seen in row (a). The original surface albedo for those frame is depicted in row (b), while the reconstructed surface albedo is shown in row (c). The bright spots on the neck, e.g., stem from a constant bright illumination in this body region. As it is not possible to directly separate material brightness from illumination, the albedo in these regions was assumed to be brighter.

The naïve approach of setting the V component to a constant value during reconstruction to compensate for too dark albedos, however, does not work since the color black requires $V = 0$ while bright colors or white require $V = 1$. Either

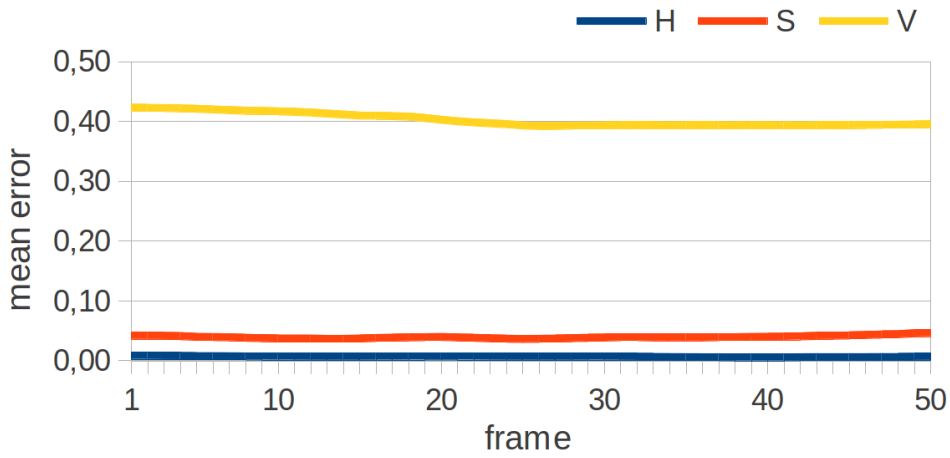

Figure 5.2: Per-channel mean error of the reconstructed HSV albedo of every surface point over the entire synthetic sequence. The separate channels hue (blue), saturation (red), and value (yellow) are reconstructed at different levels of accuracy. While hue and saturation are reconstructed very well with less than 2% resp. 5% error, the color brightness is heavily influenced by the *unknown* environmental illumination and can, therefore, only be reconstructed as a scaled version of the original material brightness. Being reconstructed with a mean error of about 40% still shows a tendency towards the correct brightness.

very dark or very bright colors are reconstructed with high error in this case. My proposed approach, on the other hand, reconstructs the value component of the surface albedo as a scaled version of the original albedo. Therefore, in the application of illumination estimation, the value component of the reconstructed albedo can be rescaled according to a constant illumination intensity, or vice versa, to remove the effect of this reconstruction error.

The proposed technique was also evaluated using three real-world sequences with a variety of illumination settings and surface materials. The scene in

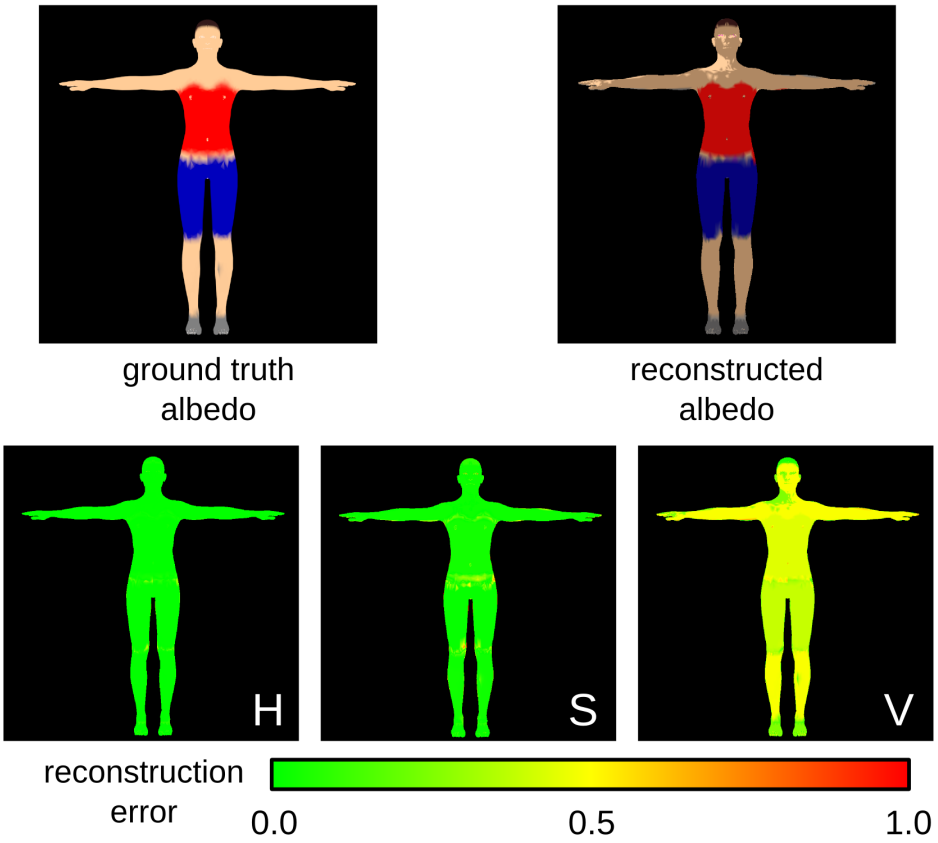

Figure 5.3: Per-channel mean error of the first frame of our synthetic sequence: Top: Ground-truth albedo (left) vs. estimated albedo (right). Bottom: The strongest deviation from the desired albedo color can be seen in the value component of the reconstructed HSV albedo color (3rd image), while the reconstruction of hue and saturation (1st and 2nd image, resp.) yield very good results.

Figure 5.4(a) was taken from the motion picture *Into The Blue*. The sequence *Haidi*, Figure 5.4(b), was taken from the *i3DPost* dataset [GKH+09; SH07], and the *Yoga* sequence used in Figure 5.4(c) was taken from a video available on online. A plausible albedo reconstruction could be obtained for every sequence,

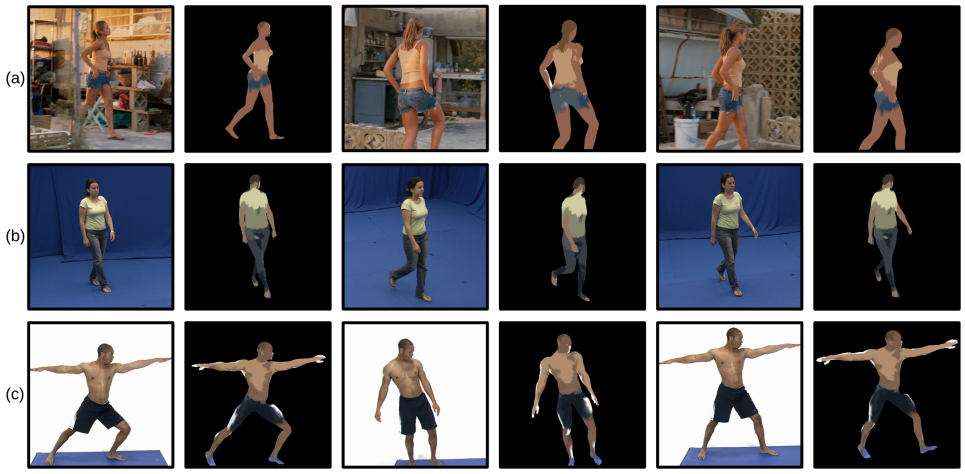

Figure 5.4: Real-world scenes used for testing: (a) taken from the motion picture *Into The Blue* ©MGM, (b) taken from the *i3DPost* dataset, and (c) taken from an online video ©Stock Footage.

matching the visual appearance of the actors. The albedo of skin, hair and clothes could be reconstructed uniformly for the *Into The Blue* sequence as the actress was walking along a curve and rotating the body relative to the light sources. This reduced the influence of self-shadowing artifacts during the albedo reconstruction. In the *Haidi* and *Yoga* sequence, a more linear movement of the actors resulted in multiple albedos, e.g, for the upper body region, as shadows affected these regions in all input frames. Distinguishing between shadow or surface texture was impossible, consequently resulting in multiple albedo colors for the same surface material. These different albedo colors, however, primarily vary in their value component, as shown in the synthetic example, Figure 5.3(V).

This approach to monocular albedo reconstruction of arbitrary objects was published in [RBM14] and was presented to an audience in the field of general image processing.

5.3 Scene Illumination Reconstruction

Having available a 3D mesh representation as well as the unshaded surface albedo for every visible surface point for each frame of the input video allows for a per-frame reconstruction of scene illumination.

By analyzing the visual difference of the unshaded object compared to the final shaded result as provided by the original input video, a set of incident illumination directions can be estimated. The animated and textured 3D geometry provides 3D position, surface orientation, and HSV surface albedo for every surface point in every frame. Assuming a Lambertian shading model [BJ03], the final visible surface color is a result from these input variables combined with the sought set of incident illumination vectors and light source colors. The shading equation

$$L_o(\mathbf{x}, \omega_o) = \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i)(\omega_i \cdot \mathbf{n}) \; d\; \omega_i \tag{5.1}$$

can be rearranged and solved for the illumination parameters of $L_i(\mathbf{x}, \omega_i)$ using the available information of the original image $L_o(\mathbf{x}, \omega_o) = \mathbf{I}_t$, the geometry $\mathbf{n}$ at each surface point provided by the results of Chapter 4, and the surface reflectance model $f_r(\mathbf{x}, \omega_i, \omega_o)$ recovered in Section 5.2.

As the problem is ambiguous, not knowing the exact number of light sources, I decided to constrain the problem by using a fixed set of L potential light

sources, as proposed by Gibson *et al.* [GHH01]. This reduces the complexity of the overall reconstruction problem. It can now be formulated as a linear system

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \tag{5.2}$$

describing the shading equations for a set of L light source candidates which can be solved in a least-squares sense. The $3L \times 1$ vector $\mathbf{x}$ is describing the RGB components of all sample light sources, $\mathbf{b}$ is the vectorized version of the original input image $\mathbf{I}_t$, and each row of the matrix $\mathbf{A}$ represents the influence of each light source to every image pixel according to the used shading model.

Using a Lambertian shading model based on the albedo estimate of Section 5.2 already reproduces good results. The approach is theoretically able to use more complex shading models based on BRDFs.

In contrast to Gibson *et al.* [GHH01] I decided to arrange the sample light sources equidistantly on a camera-oriented hemisphere using a golden spiral point distribution [SP06], Figure 5.5. This set of light sources properly describes the majority of the shaded and *visible* surface pixels, while light sources behind the object only contribute to the shading of boundary pixels. These regions on the geometry, however, provide less reliable surface orientation information due to the still imperfect shape and pose reconstruction of body model $\mathbf{B}$. Therefore, the selected set of L light source candidates allows to describe the set of *reliable* surface points optimally, while potential light sources supported only by unreliable surface sampling areas are explicitly omitted.

After empirical evaluations using $N = 400$ light sources seemed to be sufficient to describe a scene's illumination plausibly.

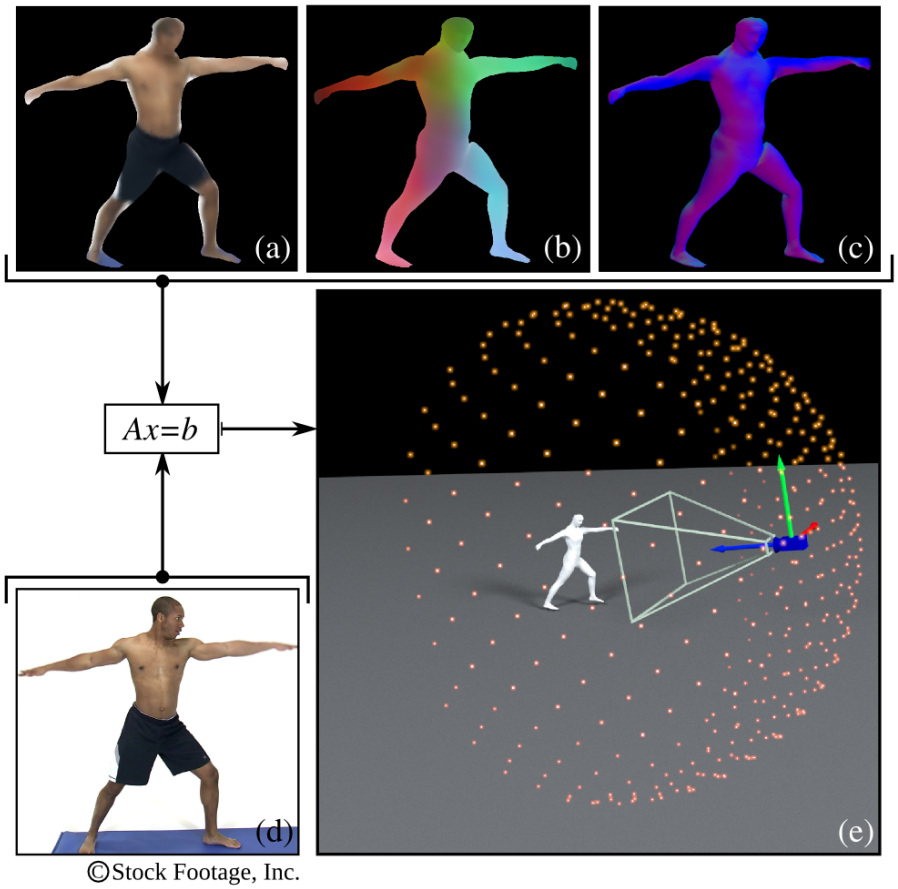

Figure 5.5: Illumination Reconstruction: The illumination is reconstructed for a fixed set of light sources, equidistantly arranged on a camera-oriented hemisphere (e). Comparing the target image (d) with the rendering using the reconstructed surface albedo (a), body geometry (b), and surface orientation (c) allows to estimate the color and brightness of the light sources.

To further reduce the complexity of the linear system, not every surface point is taken into account. The colors are sampled from the final image into $\mathbf{b}$ only for an equidistantly sampled subset of surface points. A shading equation can

be formulated according to the local surface geometry for every light source position in $\mathbf{A}$. The entirety of these sample supported equations yields a system of equations $\|\mathbf{Ax} - \mathbf{b}\|_2$ that can be solved for $\mathbf{x}$ as a minimization problem in a least squares fashion. The color difference of the shaded 3D body model and the original input image $\mathbf{I}_t$ of the selected sample positions is minimized by selecting a local minimum of active light sources in the fixed set of light source candidates. The result is a vector of light colors for all N light source candidates on the hemisphere, while all inactive light sources are assigned black, $(0, 0, 0)$. By integrating an additional constraint to the linear problem, the number of active light sources can be reduced to a minimum. This also helps to reconstruct the original scene illumination more realistically as the number of existing light sources is naturally rather small.

Using a non-negative least squares solver (NNLS) [LH95] allows to compute a physically correct solution that prevents light sources from emitting negative energy and canceling each other out in the final solution. This, again, reduces the search space of valid local minima and helps converge to a final solution very quickly.

This technique is able to estimate the incident illumination on a per-frame basis, as the color and surface samples $\mathbf{I}_t$ and $\mathbf{B}_t$ taken from the same frame in time It allows to reconstruct dynamic scene illumination over the complete sequence. Temporal filtering of the N light source colors removes high frequency fluctuations of the illumination intensity, yielding a smoothly animated scene illumination, Figure 5.6.

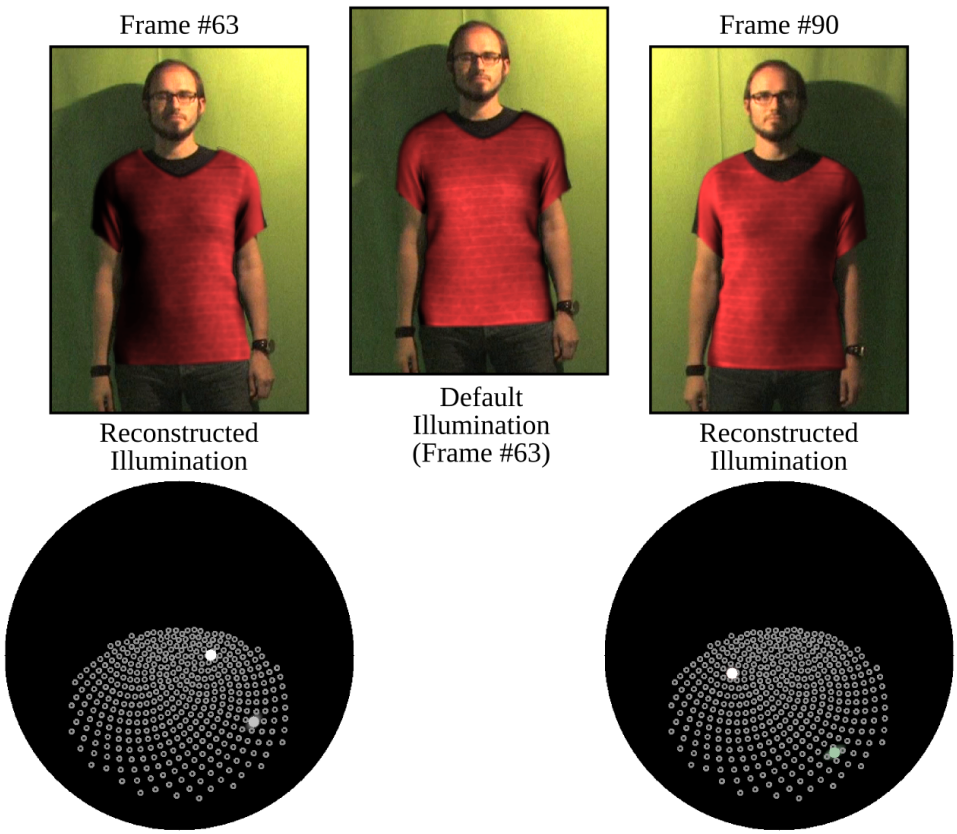

Figure 5.6: Example frames of a sequence with a moving light source: The images show the effect of the reconstructed animated light source compared to static lighting. The moving shadow in the background gives an idea of the original illumination direction. If a virtual object does not resemble the same illumination conditions, it is less probable in the composed image. The spherical illustrations on the bottom row depict the reconstructed light source positions and colors.

5.4 Discussion

The proposed approaches to albedo reconstruction in Section 5.2 and scene illumination estimation in Section 5.3 exploit the reconstructed geometry in-

formation created in Chapter 4. In contrast to other approaches [GVWT13; VWB+12], this knowledge about the geometry can be used to effectively reduce the complexity of the problem to a minimum.

Assuming a moving object, the reconstructed geometry can be used to recover a complete surface albedo map in a statistical analysis of visible surface points. This approach, however, fails for static objects in a scene with non-moving light sources as it directly makes use of changing illumination onto the object's surface stemming from object or light source motion. For the use case of actor augmentation in monocular video motion of actor, camera, or lights is a valid assumption. In case of self-shadowing that does not change over time, the surface albedo might be reconstructed slightly darker in the affected body regions. This does not affect the subsequent illumination reconstruction much as it uses the entire visible surface albedo for every potential light source.

The illumination reconstruction processes only those light source positions, that can be effectively estimated using the data available. In comparison to Gibson *et al.* [GHH01] the uniform light source distribution on a hemisphere ensures a realistic illumination reconstruction for various scenes, while the additional non-negative least squares solver prevents impossible solutions of the optimization problem. The separated estimation of albedo and illumination furthermore allows to reconstruct animated scene illumination, as the albedo information can be evaluated for every frame of the sequence. However, estimating the illumination as a combination of few, local light sources does not properly resemble light sources having covering large areas. To reconstruct this kind of illumination, a very dense set of potential light source positions need to be used for reconstruction, making the illumination reconstruction more complex.

6 Realistic Video Augmentation and Rendering

A realistic augmentation of existing video data requires virtual objects to be convincingly and credibly embedded into the original scene. Multiple factors come into play to make this embedding visually plausible and believable to a viewer.

First of all, the motion and interaction of the object must correspond to the objects originally present in the video. Interactions must result in plausible behavior and objects must be able to occlude each other. These occlusions give a lot of information about the depth ordering of the objects in the scene and their relative sizes. Object-object interactions must result in physically believable behavior. In the case of deformable virtual objects like fluids or clothing, sophisticated simulation techniques exist to create physically plausible deformations and resulting object motions. To be able to let a virtual object interact with the real objects of the original scene, these objects need to be reconstructed in 3D shape, position and motion in order to be used as constraints in physics simulations.

The virtual objects need to be rendered realistically. This requires sophisticated materials and rendering techniques, creating highly realistic looking images of the virtual object. Nowadays, state-of-the-art path tracing techniques in combination with complex material models allow to create very realistic renderings of dynamic objects and complex materials such as woven fabrics and clothes [KJM08].

As a third factor, plausible illumination is crucial to a believable embedding of virtual objects into existing imagery. The human visual system is used to physical rules like shadow casting of illuminated objects to derive additional depth cues from object shading. Wrong illumination might cause shadows to fall into an unexpected direction, resulting in an unrealistic impression to the viewer. Also illumination colors and general brightness need to match the original scene, as an object using a highly realistic material still looks artificial if the shaded surface appearance does not match the objects nearby.

6.1 Motivation

An interesting field of realistic video augmentation is *virtual clothing*. The actor is augmented with a set of virtual garments that are animated according to the actor's original motion. Having a realistically shaped and animated 3D body model for a person in a monocular video allows to use the animated geometry in a physically correct cloth simulation as a collision body for virtual clothing. The resulting animation of the virtual clothes will match the motion of the actor in the original scene and allows for realistic video augmentation. In combination

with the reconstructed, dynamic illumination model and sophisticated fabric material models, a high quality video augmentation is achievable.

6.2 Virtual Clothing and Image-based Video Compositing

A realistic clothes simulation is very important for a realistic augmentation result. The 3D body model provided by the proposed algorithms in Chapter 4 already yields a plausible representation of the actor's shape and motion. Still, the reconstructed model only approximates the actor to a certain extent. Existing clothing on the actor is not modeled by the parametrized body model. This is especially important for wide clothes such as coats, jackets, or skirts. A direct augmentation is therefore not possible as those original clothing parts will likely not be covered up by the additional virtual garment, fitted to the parametrized body model without this wide clothing. Therefore, after a physical simulation of the virtual clothing, image-based corrections have to be made to properly augment the final video.

In the following I describe a pipeline for video augmentation consisting of three basic steps. At first, the animated body model is used to drive a clothes simulation generating an animated mesh model of the virtual garment. Image-based warping techniques are then used to compensate for insufficient alignment of virtual garments, original actor silhouette, and small motion differences. In a third step, the garment is adapted to the original silhouette more precisely

using contour warping in regions where the virtual garment does not exactly match the actor silhouette.

6.2.1 Realistic Cloth Simulation

Using a sophisticated cloth simulation method, highly realistic clothes can be created. Some of these simulation tools make use of particle-based simulation of the elastic material [BHW94; BW98], that provide high quality results while still being fast enough to model and simulate dynamic materials at interactive rates [Zel05]. This interactivity is necessary to properly drape a cloth model over a human mesh model in a reference pose, apply optional fixture constraints like a virtual belt or buttons to the mesh model surface, and tweak material parameters prior to the full simulation.

Other more elaborate and advanced techniques, allow one to model and simulate virtual fabrics at yarn level [KJM08]. With these approaches, highly realistic deformations and stretching behavior can be simulated, especially for wool fabrics.

Special modeling tools have also been developed to create the underlying yarn structure automatically from a predefined pattern provided as an input mesh defining the general surface [YKJM12]. The yarn layout is then generated according to the original mesh faces, and a highly complex garment model is created.

These techniques are still very elaborate A commercial particle-based garment simulation already creates very realistic results, so I chose to use an already well-established modeling tool for a particle-based garment simulation that

was sufficient for realistic video augmentation. However, as this step is not dependent on any other part of the video augmentation pipeline, it may as well be exchanged for any other high-quality garment simulation technique. For demonstration purposes I chose to use the simulation tool *Marvellous Designer* to create highly realistic cloth animations for a given animated human mesh model [Des].

A standard rest pose (T-pose) of the body model shaped using the estimated parameters from Section 4.2 allows to model and drape the virtual clothing to optimally fit the actor in the video. Even complex clothes using multiple layers of fabric or different materials with different bending and stretching behavior can be used to model very detailed and realistic garments, as shown in Figure 6.1.

Eventually, the reconstructed animation from Section 4.3 is used to drive the cloth simulation and generate a plausible garment animation matching the motion of the actor. The animated mesh model serves as the collision body for all simulated, dynamic objects. As the body model is an approximation to the original actor using as much detail as possible, even complex interactions of actor and virtual garment are possible to simulate, Figure 6.2.

The simulation result is an animated mesh model of the virtual garment that shows realistic bending, stretching, and folding behavior according to the original motion of the actor in the input video.

The animated cloth model can be rendered using high-quality materials using the reconstructed scene illumination of Chapter 5. The path tracer *VRay* yielded realistic-looking results, see Figure 6.3.

Figure 6.1: Garment Examples: Displayed are three different garment models animated using a realistic cloth simulation. All garments were designed by Michael Stengel, colleague and PhD student at the computer graphics lab. The dress in (a) was used for the test sequence *Ballet*, the shirt (b) in the sequence *Yoga* and the more complex dress in (c) was used for augmenting the *Into The Blue* test sequence.

6.2.2 Alignment Correction

This first animation of the virtual garment model is created corresponding to the motion of the animated body model. However, this animation and the animated body geometry itself stem from the non-exact 3D reconstruction, Chapter 4, which still lacks fine surface details and motion of small detailed body parts. This imprecision is system-inherent as the parametrized body model is not able to describe arbitrarily fine details or actor-specific details, no hair, and also no additional clothing. Therefore, the animated garment does not perfectly match the motion of the *original* person in the video. The animation and motion is

Figure 6.2: Illustration of interaction between actor and virtual object: The grey jacket in the exemplar augmentation result above is virtually created and blended into the original image. As the body model properly resembles the original actor, interactions between virtual garment and actor are possible.

based on the *naked* body model, but not the actual appearance of the actor in the original input video sequence. Also, during the motion reconstruction in Chapter 4, a small lag may have been introduced by the temporally filtered smoothed motion. This deviation from the original actor motion can be visually confusing, even if it is spatially small. The human eye is very sensitive to motion and this impairs the quality and credibility of the direct video augmentation. To increase realism and to improve the final augmentation result, the garment has to be better aligned to the actor's motion.

The first step towards a motion-corrected garment animation is the precise motion alignment of the garment to the whole body motion. By analyzing the

Figure 6.3: Garment Rendering under Scene Illumination: According to the pose of the actress in (a), the virtual dress was animated and rendered using high-quality materials (b) and reconstructed scene illumination (c) to achieve a realistic augmentation result (d).

screen-space motion vector fields of the original video and the reconstructed body model, the motion difference can be reduced by finding a correspondence field of actor and garment motion and applying the inverse of the per-pixel motion difference to the rendered garment in form of image warping. To avoid tearing of the virtual garment, the difference of the motion vector field is smoothed locally prior to the warping of the rendered cloth.

In the proposed approach, the user needs to initialize the correspondence field with a manually selected ideal cloth fit to allow for feature matching of actor and garment. This is done by selecting a small set of key frames describing an optimal fit of the virtual garment. A set of ten or less key frames has been sufficient in all test sequences. Feature correspondences established this way can now be used to identify and track motion misalignments. The 2D image

space motion of the reconstructed body model is known, and can be directly computed in the rendering step. However, the corresponding motion of the actor has to be recovered from the input video data. Using a long-range optical flow based on belief propagation and SIFT feature matching [LLN+10], a continuous motion vector field can be generated for the actor of the original input sequence. This technique is able to compute per-pixel motion vectors for every frame t that can be accumulated to describe image space motion trajectories $\mathbf{T_I}(x, y, t)$ for each pixel (x, y). Similar trajectories $\mathbf{T_B}(x, y, t)$ can be established for every pixel of the corresponding body model.

These trajectories $\mathbf{T}(x, y, t)$ describe the displacement of a pixel (x, y) for an intermediate frame t relative to a previous key frame. The comparison of the per-pixel trajectories of original actor and body model describe the difference of the body model's motion relative to the original actor motion. Subtracting $\mathbf{T_B}$ and $\mathbf{T_I}$ from each other and storing this difference vector in a difference map $\mathbf{D}$ at the warped pixel position of $\mathbf{T_I}$ for every frame yields a correction vector for every pixel in every frame

$$\mathbf{D}(x', y', t) = \mathbf{T_I}(x, y, t) - \mathbf{T_B}(x, y, t), \tag{6.1}$$

where (x, y) is a pixel position in the key frame and (x', y') is the pixel position according to $\mathbf{T_B}(x, y, t)$, Figure 6.4.

As the motion of the actor in the original sequence and also the reconstructed motion of the body model are assumed to be locally smooth, the correction vector field for every frame should be locally smooth as well. This allows to

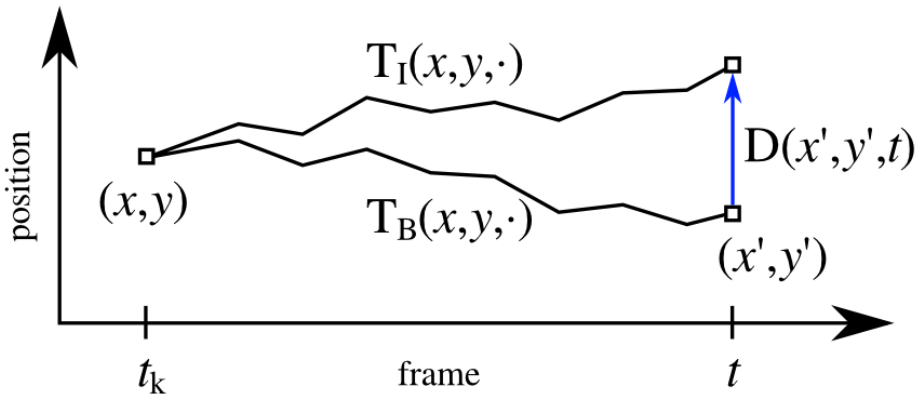

Figure 6.4: Motion correction from pixel trajectories: Accumulated trajectory of a pixel in the original input image and its correspondence on the reconstructed body model. Starting from a user-defined key frame, the slightly different motion vectors in each image accumulate to a perceivable spatial disparity. The difference vector for every intermediate frame can be used to compensate for this motion divergence.

remove outliers by applying a small median filter to the generated motion correction map $\mathbf{D}$.

Due to possible occlusions, trajectories may not exist for every pixel in every frame. This missing information of a motion correction vector $\mathbf{D}(x, y, t)$ to a pixel (x, y) at an intermediate frame t can be interpolated from neighboring information using an edge-preserving domain transform filter [GO11]. This allows one to create a fully populated and locally smooth motion correction map for every intermediate frame. The per-pixel motion correction vectors $\mathbf{D}(x, y, t)$ can then be applied to the rendering of the virtual garment $\mathbf{G}_t$, compensating for the motion difference of original actor and reconstructed body motion.

6.2.3 Silhouette Warping

After correcting the motion vectors of the virtual garment to the original actor motion, the final composite can still suffer from some silhouette alignment inconsistencies. As the parametrized 3D body model used for estimating and reconstructing the actor's body shape in Section 4.2 is not able to describe every fine body detail and cannot describe any clothing the actor wore in the original video sequence, the silhouette of the reconstructed body model and the original actor silhouette still differ. To compensate for this misalignment, for every frame t a silhouette contour warping is computed from the actor's original silhouette $S_{\mathbf{M}_t}$ and the artificial garment silhouette $S_{\mathbf{G}_t}$ in those regions where the artificial garment silhouette does not overlap the original actor silhouette appropriately, Figure 6.5.

In a semi-automatic process, the user initializes the most significant silhouette sections by roughly matching contour segments in both silhouettes for a small set of key frames. Depending on the actor's motion this can vary between one to ten percent of the input sequence. By defining the start and end points s_{start} and s_{end} of a silhouette segment on both silhouettes of the virtual garment $\mathbf{G}_{t_0}$ and of the actor $\mathbf{M}_{t_0}$ in a key frame t_0, the silhouette segments $S_{\mathbf{G}_{t_0}}$ and $S_{\mathbf{M}_{t_0}}$ can be constructed by tracing the silhouette contour. Modeling both silhouette contour segments as parametric curves allows to match points on either contour segment with the same interpolation parameter. In a subsequent key frame t_1, preferably at the end of the sequence, the user again selects the start and end points of the silhouette contour segments. Using linear interpolation and snapping the interpolated start and end points to the closest contour pixel, for every

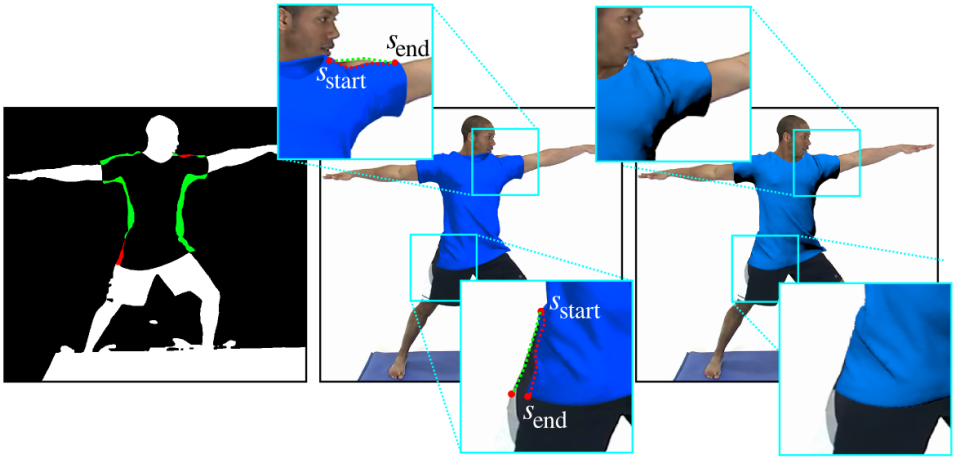

Figure 6.5: To align the rendered garment with the actor in the video (©Stock Footage) more accurately, we warp silhouette mismatches in image space. (a) Video frame and rendered garment with mismatching silhouettes (in green and red). (b) Composite without silhouette warp. (c) Composite after image-space refinement and depth-based compositing.

intermediate frame a traced contour segment $S_{\mathbf{G}_t}$ for the garment silhouette and $S_{\mathbf{M}_t}$ for the actor silhouette between s_{start} and s_{end} can be computed for all frames t with $t_0 \leq t \leq t_1$. Using the interpolation parameter of both curves, a full curve matching can be generated for every contour segment. As the morphology of the actor silhouette might change over time, additional manually defined keyframes inbetween t_0 and t_1 help to resolve matching ambiguities.

To let the contour warp derived from the curve matching affect also neighboring regions, an image-based diffusion using radial basis functions is emnployed to create a smooth warp field for contour neighborhoods in every frame. The final warp field is then used to deform the contour of the virtual garment according to the original silhouette in the regions of false overlapping.

For simplicity, the per pixel warp fields for motion correction (Section 6.2.2) and silhouette mismatch (Section 6.2.3) in image space can be combined and applied together in a single step to the original garment rendering $\mathbf{G}_t$. The result is a slightly warped garment rendering $\mathbf{G}'_t$ where the motion mismatch and inconsistent silhouettes due to missing geometry detail have been reduced to a minimum. This step is important to preserve visual plausibility of the final augmentation result.

6.3 Results

The complete set of tools described in Chapter 4, Chapter 5, and Chapter 6 allows one to create a fully integrated and easy-to-use video augmentation pipeline, Figure 6.6, that was published in [RKS+14]. For a realistic video augmentation, several problems have to be solved. A realistic actor representation has to be derived from the monocular video information. Secondly, the scene illumination has to be reconstructed in a realistic way, as wrong shading heavily affects augmentation quality. Lastly, the garment has to be rendered and animated realistically before being blended into the source video.

Starting from a monocular input video, the user defines an initial pose of a parameterized and deformable human body model. The shape optimization then recovers the actor's approximate body shape, and the pose reconstruction uses this information to recover the actor's motion in the most plausible way. Small user-constraints may help to remove possible ambiguities during motion reconstruction. The animated body model is then used to drive a highly realistic cloth simulation, creating a properly animated mesh model of an artificial piece

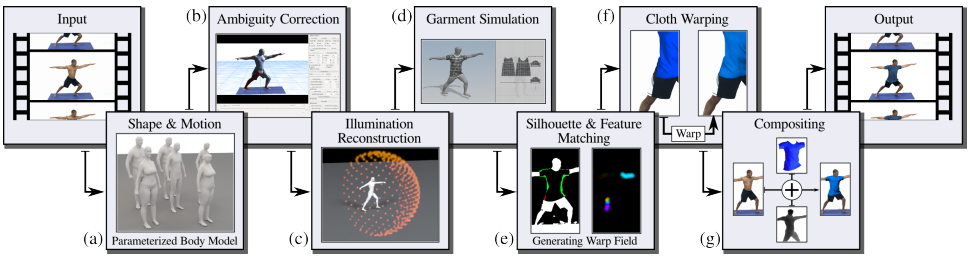

Figure 6.6: Overview of the Video Augmentation Approach: Starting from an arbitrary uncalibrated input video, at first the silhouette mattes are extracted (a) and used to optimally fit the shape and pose of a parameterized body model (b). The reconstructed 3D geometry data of the body model allows one to reconstruct a dynamic scene illumination (c). The 3D model is also used to animate a virtual garment (d) using a high-quality cloth simulation. Silhouette analysis (e) and image space motion detection (f) help to properly align the virtual garment with the original actor. In a final compositing step (g), the animated garment is rendered using highly realistic materials under the reconstructed dynamic illumination and blended into the original video stream. As a result the actor in the video is realistically augmented with artificial clothing.

of clothing. To reconstruct original scene lighting, the animated body model also helps to estimate the most plausible light source positions and their intensities per frame. This way a fully dynamic illumination reconstruction is possible for the complete video sequence which allows to further improve the desired realism of the augmentation. To compensate for alignment errors of the virtual garment relative to the actor, it is also necessary to remove motion artifacts induced by imprecise reconstruction of the body model in shape and motion. This is inevitable as the parameterized body model is not able to describe a person's clothing in every fine detail, but only yields a good approximation to the naked body shape. These shape differences cause small misalignments of original actor

and virtual garments that can be effectively reduced by image-based motion correction. The animated garment is compared against the silhouette and motion of the original actor, allowing to correct for these minor misalignments in an image-based alignment correction step based on image warping. The animated garment is finally rendered with a state-of-the-art path tracer using realistic fabric materials and the reconstructed dynamic illumination information. Using visibility and self-occlusion information of the 3D body model, the rendered garment can be properly blended into the original video data. This creates a realistic impression of the actor wearing the virtual garment.

Using a set of eight different test sequences with a variety of actor shapes, motion complexity, and scene illumination conditions, the presented augmentation system was be evaluated, Table 6.1. Exemplar augmentation results can be seen in Figure 6.7 - Figure 6.14.

Scene	Frames	Edited frames		Editing time (in min)		Total processing time (in min)
		pose	warp	pose	warp	
Ballet	93	25	81	20	60	360
Dancer	140	27	89	20	50	1200
Dynamic Light	240	7	66	5	82	820
Haidi	110	20	79	20	40	420
Hulk	150	9	0	6	0	760
Yoga	299	12	139	15	48	1380
Into The Blue	149	22	7	25	3	250
Parkour	256	35	12	30	10	420

Table 6.1: The first two columns state the test sequence name and the total number of frames of the individual sequence. In column 3 the number of manually edited frames for body pose and image warp corrections is listed. The number of edited frames regarding corrections for pose and motion (Section 4.3) depends on individual complexity of the motion performed by the actor in the sequence. The number of frames with image-based corrections for motion and silhouette alignment is the total of all key frame editing operations required for the warp computation in Section 6.2.3. The corresponding total of manual time spent for key frame-based editing operations is listed in column 4. Column 5 presents the computer processing time of a CPU implementation on a commodity PC (Intel Core i7, with 3.2 GHz and 12GB RAM).

Figure 6.7: Original (left) and augmented frames (right) of the test sequence *Ballet* ©Howcast Media taken from an online video featuring fast rotational motions of the dancer, [How11].

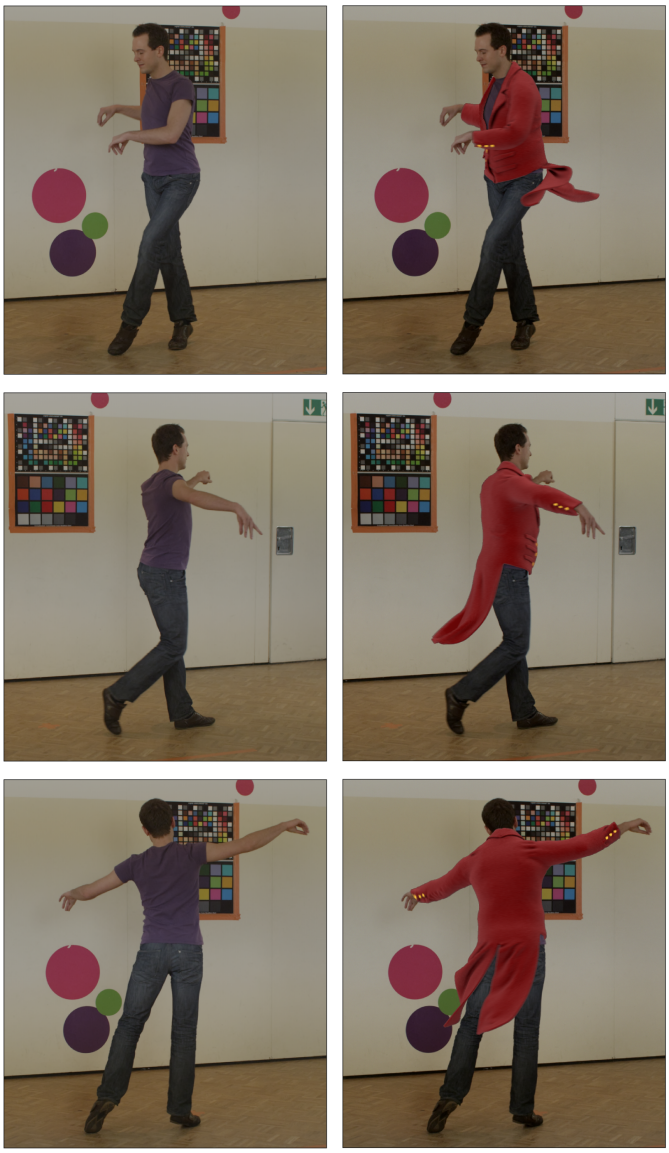

Figure 6.8: Original (left) and augmented frames (right) of the test sequence *Dancer*. The scene was used to test the reconstruction of a combination of slow and fast motions of the dancer.

Figure 6.9: Original (left) and augmented frames (right) of the *Dynamic Light* test sequence featuring a bright moving light sources. This scene was especially used to evaluate the capability of the illumination reconstruction approach, being able to reconstruct a dynamic scene illumination.

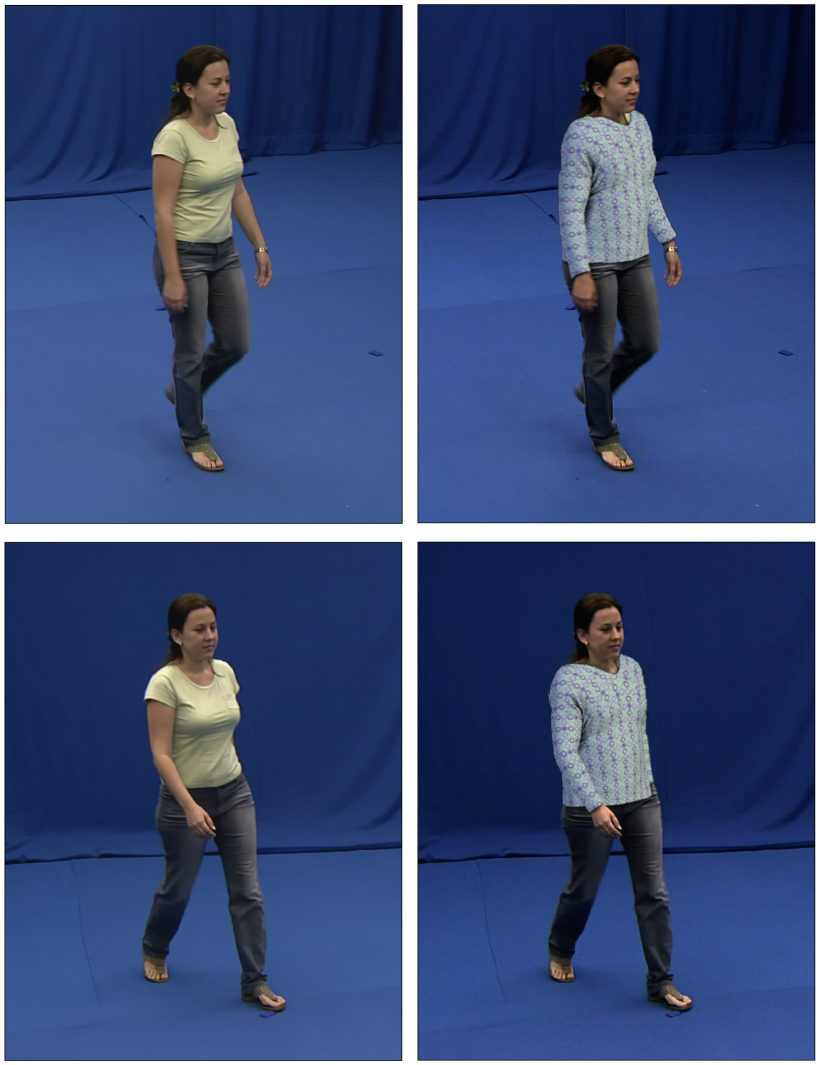

Figure 6.10: Original (left) and augmented frames (right) of the test sequence *Haidi* taken from the *i3DPost* dataset [GKH+09].

Figure 6.11: Original (left) and augmented frames (right) of the *Hulk* test sequence. This video was used to demonstrate the possibility to interact with virtual objects, as the actor in the video is realistically ripping of his artificial shirt.

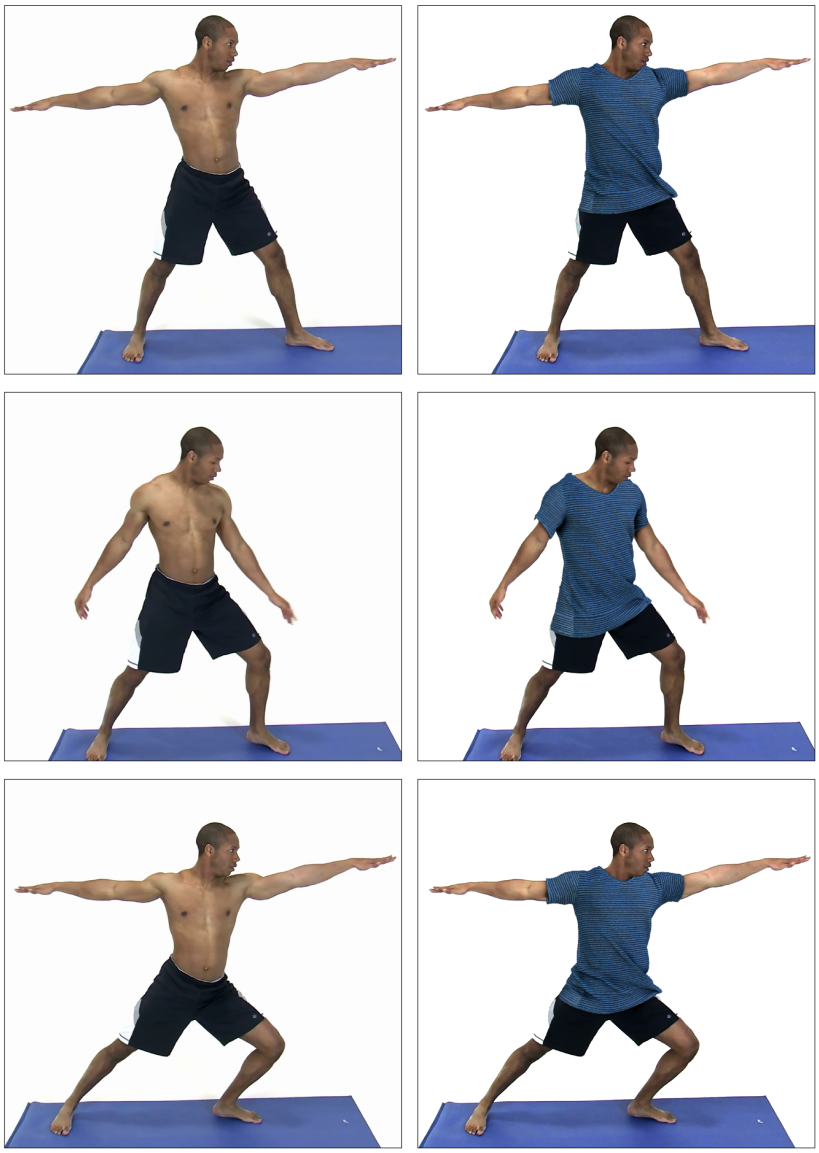

Figure 6.12: Original (left) and augmented frames (right) of the test sequence *Yoga* ©Stock Footage [Sto11]. It was chosen to demonstrate the realistic and slow deformations of the simulated garment. Especially the slow-moving objects must be properly aligned to the actor's motion in order to be credible in the composed video.

Figure 6.13: Original (left) and augmented frames (right) of the test sequence taken from the movie *Into The Blue* ©MGM [MGM05]. The scene shows the capability of realistically augmenting existing video footage, as the sequence was taken from an eponymous motion picture.

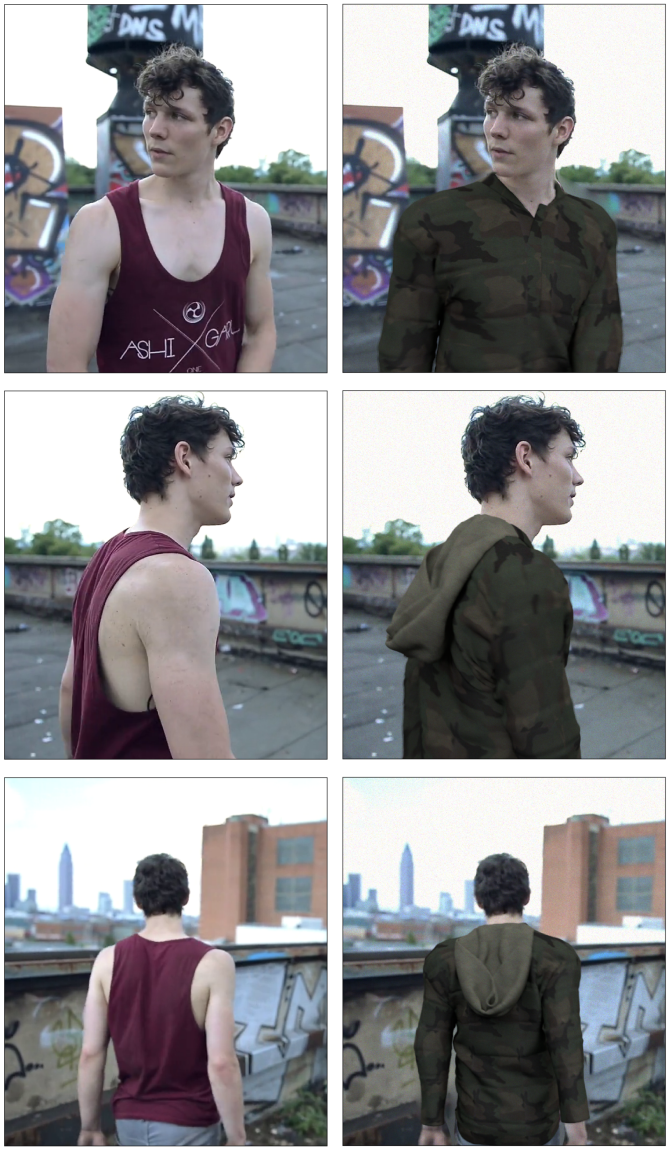

Figure 6.14: Original (left) and augmented frames (right) of the *Parkour* test sequence that was taken from the short film *Aaron Martin - Parkour* provided by Yannick Wolff [Wol13].

The sequence *Ballet*, Figure 6.7, depicts a ballet dancer performing a fast pirouette motion, while the *Dancer*, Figure 6.8, scene contains a combination of slow and fast movements. The sequence *Hulk* allows to evaluate complex interactions of the actor and virtual garment, Figure 6.11, while the *Yoga* sequence depicts time-coherent realistic folding of clothes, Figure 6.12. The slow motion of the garments in both sequences are a good validation of the image-based motion correction, as especially for slow moving objects correctly aligned motion is important for the credibility of the virtual object. The *Dynamic Light*, Figure 6.9, sequence was used as an example for a scene with a moving light source and to evaluate the capability of the per-frame illumination reconstruction technique described in Chapter 5. While the scenes *Ballet* and *Yoga* are taken from online videos [How11; Sto11], *Dancer*, *Dynamic Light*, and *Hulk* are own recordings. The *Haidi* sequence is taken from the *i3DPost* dataset provided by Gkalelis *et al.* [GKH+09; SH07] and features a straight walking motion, Figure 6.10. The sequence *Into The Blue*, Figure 6.13, is a short clip taken from the eponymous movie [MGM05] and the sequence *Parkour*, Figure 6.14, is a section of the short film *Aaron Martin - Parkour* provided by Yannick Wolff [Wol13]. Both clips demonstrate the applicability of the proposed approach to professional recordings with strong, continuous camera motion. All scenes have a resolution of 1080p except for *Ballet* (720p) and *Dancer* (4k).

The number of manually edited frames during shape and pose optimization, cf. Table 6.1, includes the initial positioning of the body model for shape reconstruction as well as the corrections of falsely reconstructed joint orientations after automatic pose reconstruction. The amount of necessary manual guidance

of the body and silhouette matching algorithm (Section 6.2.2) is included in the 3rd column of Table 6.1. Scenes with more complex or fast motions required more corrections of the pose estimation (*Ballet*, *Dancer*) than scenes with slower motions (*Dynamic Light*, *Yoga*). Pose correction was necessary in frames with ambiguous silhouettes resulting from body self-occlusions in the scenes *Haidi*, *Hulk*, *Into The Blue*, and *Parkour*. Editing a pose with the proposed key frame system was not very time consuming and usually took less than a minute per edited frame, in general.

User guidance for image-based refinements was needed in those frames containing strong silhouette deformations. A silhouette mismatch could be corrected in a single frame within a few seconds, and since the silhouette start and end position are interpolated linearly, corrections for slow moving objects were completed quickly. For faster and non-linear motions, however, more editing time is required. The overall quality of most sequences could be increased considerably by applying a few corrections, and the majority of the time listed in the fourth column of Table 6.1 was spent on correcting small details.

On average, the total manual interaction time using the provided tools took approximately one minute per edited frame. The total editing time for an entire sequence varied between 60 and 90 minutes. This amount of manual interaction is still permissible to achieve a highly realistic augmentation, as state-of-the-art approaches allow comparable amounts of user guidance, being in the range of one to ten minutes per edited frame [JSMH12].

6.4 Discussion

The presented strategy for video augmentation of actors in arbitrary monocular video sequences shows that it is possible to create highly realistic augmentations with a minimum of user interactions. The estimated body shape and motion from monocular video, Chapter 4, and the estimated dynamic scene illumination from Chapter 5 allow one to create realistically animated and shaded virtual garments. However, due to the inability of the parameterized body model to describe clothing or hair, the reconstruction body shape still lacks detail and only imperfectly matches the original actor's appearance. In contrast to Jain *et al.* [JSMH12], who use the body reconstruction for image-warping purposes, misalignments of body model and actor are more obvious in a video augmentation application. To compensate for the alignment error, an additional image-based correction step is necessary. Using only few user constraints, a pixel precise alignment and motion correction can be computed. This heavily affects the final augmentation result as floating artifacts of the virtual garment are reduced and it's contour matches the actor silhouette.

Warping the garment contour to match the original silhouette, however, fails for wide clothing worn by the actor. The virtual garment cannot be distorted much while preserving believable shape and motion. The solution to this problem is to remove excess clothing from the original video frames by employing image-based inpainting techniques. Still, for most videos the proposed system yields plausible augmentation results without any additional cloth removal.

7 Conclusion

In this thesis I have presented two different approaches towards an image-based augmentation of an actor from monocular input data for the application of virtual clothing. While the former approach aims at a fast and robust pose estimation using a marker suit and a novel pose descriptor that incorporates information about marker neighborhoods and local image gradients, the latter focuses on realistic video augmentation as a post-process. Both techniques allow reconstructing the pose and motion of an actor from monocular videos.

The pose reconstruction on single images, applicable at real-time frame rates for monocular video data, makes the approach described in Chapter 3 suitable for virtual try-on systems. However, the proposed technique relies heavily on a marker suit and is specifically geared towards the used marker pattern. This prevents the technique to be usable for arbitrary monocular input data or existing videos. Also, the method does not allow one to reconstruct a highly realistic body model, only the skeletal motion. In a virtual mirror application, a generic body proxy has to be used instead of a body model that optimally matches the actor's shape.

To tackle these limitations, the second approach focuses on the realistic reconstruction of body shape and motion, Chapter 4, and scene illumination,

Chapter 5. The proposed video augmentation technique of Chapter 6 is based on the realistic reconstruction of actor and illumination as a sequence of video post-processing steps. It is not real-time capable, but allows reconstructing a highly realistic body shape, motion, and a realistic and animated scene illumination. These two additional components play a crucial role when it comes to the *realistic* embedding of virtual objects into a real-world scene. In the example of virtual clothing, I was able to show that even complex garments could be animated and aligned with the actor while being rendered realistically with highly accurate fabric textures under the estimated scene illumination. Image-based motion alignment corrections helped to further optimize the composition of original video data and artificial garment. The proposed pipeline allows augmenting actors in arbitrary monocular video sequences with artificial clothing. This is a big advantage over systems that require controlled motion capture setup or multi-view input data as it can by applied to any existing video as a post-process. Even the augmentation of average-quality videos taken from *YouTube*™ or *Vimeo*™ was possible, as shown in the experiments. This opens up a new field of video augmentation, as existing recordings may now be altered to exchange existing clothing, sanitize nudity in a non-distracting way, or allow for creating virtual try-ons for garment prototypes in development. By allowing for minimal user-input, the system is easy to use and enables the user to correct for reconstruction errors during the optimization process as these may occur and stem from ambiguities present in monocular source footage.

Bibliography

[ASK+05] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. „SCAPE: shape completion and animation of people“. In: *ACM Trans. Graph.* 24 (3 July 2005), pp. 408–416. ISSN: 0730-0301.

[AST+08] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. „Performance capture from sparse multi-view video“. In: *ACM Trans.Graph.* 27.3 (2008), 98:1–98:10.

[AT04] A. Agarwal and B. Triggs. „3D Human Pose from Silhouettes by Relevance Vector Regression“. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 2 (2004), pp. 882–888. ISSN: 1063-6919.

[AT06] A. Agarwal and B. Triggs. „Recovering 3D Human Pose from Monocular Images“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.1 (Jan. 2006), pp. 44–58.

[BAC09] Soma Biswas, Gaurav Aggarwal, and Rama Chellappa. „Robust estimation of albedo for illumination-invariant matching and shape

recovery“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2009), pp. 884–899.

[BB08] A.O. Bălan and M.J. Black. „The Naked Truth: Estimating Body Shape Under Clothing“. In: *European Conference on Computer Vision: Part II*. 2008, pp. 15–29.

[BBHS07] A.O. Bălan, M.J. Black, H. Haussecker, and L. Sigal. „Shining a Light on Human Pose: On Shadows, Shading and the Estimation of Pose and Shape“. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. Oct. 2007, pp. 1 –8.

[BC10] Soma Biswas and Rama Chellappa. „Pose-robust albedo estimation from a single image“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 2683–2690.

[BHW94] David E Breen, Donald H House, and Michael J Wozny. „A particle-based model for simulating the draping behavior of woven cloth“. In: *Textile Research Journal* 64.11 (1994), pp. 663–685.

[BJ03] Ronen Basri and David W Jacobs. „Lambertian reflectance and linear subspaces“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.2 (2003), pp. 218–233.

[BJK07] R. Basri, D. Jacobs, and I. Kemelmacher. „Photometric stereo with general, unknown lighting“. In: *International Journal of Computer Vision* 72.3 (2007), pp. 239–257.

[BM98] C. Bregler and J. Malik. „Tracking people with twists and exponential maps“. In: *Computer Vision and Pattern Recognition,*

1998. Proceedings. 1998 IEEE Computer Society Conference on. June 1998, pp. 8 –15.

[BMP00] Serge Belongie, Jitendra Malik, and Jan Puzicha. „Shape context: A new descriptor for shape matching and object recognition“. In: *NIPS*. Vol. 2. 2000, p. 3.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. „Shape Matching and Object Recognition Using Shape Contexts“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.4 (Apr. 2002), pp. 509–522.

[BSB+07] A.O. Bălan, L. Sigal, M.J. Black, J.E. Davis, and H.W. Haussecker. „Detailed Human Shape and Pose from Images“. In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. June 2007, pp. 1 –8.

[BSGF10] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. „The generalized patchmatch correspondence algorithm“. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 29–43.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. „Fast Approximate Energy Minimization via Graph Cuts“. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23.11 (2001), pp. 1222–1239.

[BW98] David Baraff and Andrew Witkin. „Large steps in cloth simulation“. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM. 1998, pp. 43–54.

[BWSS09] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. „Video SnapCut: robust video object cutout using localized classifiers“. In: *ACM Trans. Graph.* 28.3 (2009), 70:1–70:11.

[BYS07] A. Bissacco, Ming-Hsuan Yang, and S. Soatto. „Fast Human Pose Estimation using Appearance and Motion via Multi-Dimensional Boosting Regression“. In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on.* June 2007, pp. 1 –8.

[CBGM02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. „Blobworld: Image segmentation using expectation-maximization and its application to image querying“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), pp. 1026–1038. ISSN: 0162-8828.

[CCBK06] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. „Bilayer Segmentation of Live Video“. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on.* Vol. 1. June 2006, pp. 53 –60.

[CCC08] Tse-Wei Chen, Yi-Ling Chen, and Shao-Yi Chien. „Fast image segmentation based on K-Means clustering with histograms in HSV color space“. In: *IEEE 10th Workshop on Multimedia Signal Processing.* IEEE. 2008, pp. 322–325.

[CL04] Baisheng Chen and Yunqi Lei. „Indoor and outdoor people detection and shadow suppression by exploiting HSV color informa-

tion“. In: *The Fourth International Conference on Computer and Information Technology*. IEEE. 2004, pp. 137–142.

[CTMS03] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. „Free-viewpoint video of human actors“. In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), pp. 569–577.

[CWJ11] Xiaowu Chen, Ke Wang, and Xin Jin. „Single Image Based Illumination Estimation for Lighting Virtual Object in Real Scene“. In: *Comp.-Aided Design and Comp. Graph.* 2011, pp. 450 –455.

[Deb98] Paul Debevec. „Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography“. In: *Conference on Computer graphics and interactive techniques*. SIGGRAPH '98. 1998, pp. 189–198.

[Des] Marvelous Designer. *Marvelous Designer 4*. Website. `http://www.marvelousdesigner.com`, visited in May 2015.

[DT05] N. Dalal and B. Triggs. „Histograms of Oriented Gradients for Human Detection“. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* 1 (2005), pp. 886–893.

[DTE+04] A. Divivier, R. Trieb, A. Ebert, H. Hagen, Clemens Groß, Arnulph Fuhrmann, Volker Luckas, José L. Encarnação, E. Kirchdörfer, M. Rupp, S. Vieth, Stefan Kimmerle, Michael Keckeisen, Markus Wacker, Wolfgang Straßer, Mirko Sattler, and Ralf Sar. „Virtual

Try-On: Topics in Realistic, Individualized Dressing in Virtual Reality“. In: *Virtual and Augmented Reality Status*. 2004, pp. 1–17.

[EESM10] Martin Eisemann, Elmar Eisemann, Hans-Peter Seidel, and Marcus Magnor. „Photo zoom: High resolution from unordered image collections“. In: *Proceedings of Graphics Interface 2010*. Canadian Information Processing Society. 2010, pp. 71–78.

[EFR08] Peter Eisert, Philipp Fechteler, and Jürgen Rurainsky. „3-D tracking of shoes for virtual mirror applications“. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–6.

[EG09] Markus Enzweiler and Dariu M Gavrila. „Monocular pedestrian detection: Survey and experiments“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.12 (2009), pp. 2179–2195.

[EM10] Martin Eisemann and Marcus Magnor. „ZIPMAPS: Zoom-Into-Parts Texture Maps“. In: *Proc. Vision, Modeling and Visualization (VMV) 2010*. Siegen, Germany, Nov. 2010, pp. 291–297.

[FH04] P.F. Felzenszwalb and D.P. Huttenlocher. „Efficient graph-based image segmentation“. In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181. ISSN: 0920-5691.

[Fit] Fitnect. *Fitnect, Interactive Kft.* Website. `http://www.fitnect.hu`, visited in Jan. 2013.

[FKGK05] J. Frahm, K. Koeser, D. Grest, and R. Koch. „Markerless Augmented Reality with Light Source Estimation for Direct Illumination“. In: *Conference on Visual Media Production*. 2005, pp. 211–220.

[FL95] Pascal Fua and Yvan G Leclerc. „Object-centered surface reconstruction: Combining multi-image stereo and shading“. In: *International Journal of Computer Vision* 16.1 (1995), pp. 35–56.

[Fos14] Jeff Foster. *The green screen handbook: real-world production techniques*. CRC Press, 2014.

[FYK10] Wei-Wen Feng, Yizhou Yu, and Byung-Uck Kim. „A deformation transformer for real-time cloth animation“. In: *ACM Trans. Graph.* 29.4 (July 2010), 108:1–108:9. ISSN: 0730-0301.

[GCH+12] Stevie Giovanni, YeunChul Choi, Jay Huang, EngTat Khoo, and KangKang Yin. „Virtual Try-On Using Kinect and HD Camera“. In: *Motion in Games*. Ed. by Marcelo Kallmann and Kostas Bekris. Vol. 7660. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 55–65.

[Gel08] Tom Geller. „Overcoming the uncanny valley.“ In: *IEEE Computer Graphics and Applications* 28.4 (2008), pp. 11–17.

[GFB10] Peng Guan, Oren Freifeld, and MichaelJ. Black. „A 2D Human Body Model Dressed in Eigen Clothing“. In: *European Conference on Computer Vision*. Vol. 6311. Lecture Notes in Computer Science. 2010, pp. 285–298.

[GHH01] Simon Gibson, Toby Howard, and Roger Hubbold. „Flexible Image-Based Photometric Reconstruction using Virtual Light Sources“. In: *Computer Graphics Forum* 20.3 (2001), pp. 203–214. ISSN: 1467-8659.

[GKB03] Igor Guskov, Sergey Klibanov, and Benjamin Bryant. „Trackable surfaces“. In: *ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2003, pp. 251–257.

[GKH+09] N. Gkalelis, Hansung Kim, A. Hilton, N. Nikolaidis, and I. Pitas. „The i3DPost Multi-View and 3D Human Action/Interaction Database“. In: *Conference for Visual Media Production*. 2009, pp. 159 –168.

[GKT+12] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. „Background inpainting for videos with dynamic objects and a free-moving camera“. In: *Proceedings of the 12th European conference on Computer Vision - Volume Part I*. ECCV'12. 2012, pp. 682–695.

[GO11] Eduardo S. L. Gastal and Manuel M. Oliveira. „Domain Transform for Edge-Aware Image and Video Processing“. In: *ACM Trans. Graph.* 30.4 (2011), 69:1–69:12.

[GRH+12] P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. J. Black. „DRAPE: DRessing Any PErson“. In: *ACM Trans. on Graph.* 31.4 (2012), 35:1–35:10.

[GS08] Philip Geismann and Georg Schneider. „A two-staged approach to vision-based pedestrian recognition using Haar and HOG features“. In: *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE. 2008, pp. 554–559.

[Gun98] Steve R. Gunn. „Support vector machines for classification and regression“. In: *ISIS technical report* 14 (1998).

[GVWT13] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. „Reconstructing Detailed Dynamic Face Geometry from Monocular Video“. In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 158.

[GWBB09] P. Guan, A. Weiss, A.O. Balan, and M.J. Black. „Estimating human shape and pose from a single image“. In: *Computer Vision, 2009 IEEE 12th International Conference on*. Vol. 29. 2. Oct. 2009, pp. 1381 –1388.

[HE07] Anna Hilsmann and Peter Eisert. „Deformable object tracking using optical flow constraints“. In: *Visual Media Production, 2007. IETCVMP. 4th European Conference on*. Nov. 2007, pp. 1 –8.

[HE08] Anna Hilsmann and Peter Eisert. „Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences“. In: *Computer Vision and Pattern Recognition Workshop* 0 (2008), pp. 1–6.

[HE09] Anna Hilsmann and Peter Eisert. „Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications“. In: *Proceedings of*

the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques. MIRAGE '09. Springer-Verlag, 2009, pp. 94–105. ISBN: 978-3-642-01810-7.

[HE12] Anna Hilsmann and Peter Eisert. „Image-based Animation of Clothes". In: *Eurographics (Short Papers)*. Eurographics Association, 2012, pp. 69–72.

[HFE13] A. Hilsmann, P. Fechteler, and P. Eisert. „Pose Space Image Based Rendering". In: vol. 32. 2pt3. Blackwell Publishing Ltd, 2013, pp. 265–274.

[HRT+09] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. „Markerless Motion Capture with unsynchronized moving cameras". In: *Computer Vision and Pattern Recognition*. 2009, pp. 224 –231.

[HS85] R.M. Haralick and L.G. Shapiro. „Image segmentation techniques". In: *Computer vision, graphics, and image processing* 29.1 (1985), pp. 100–132. ISSN: 0734-189X.

[HSR+09] Nils Hasler, Carsten Stoll, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. „Estimating body shape of dressed humans". In: *Computers & Graphics* 33.3 (2009), pp. 211–216.

[HSR11] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. „Free viewpoint virtual try-on with commodity depth cameras". In: *Virtual Reality Continuum and Its Applications in Industry*. 2011, pp. 23–30.

[HSR13] S. Hauswiesner, M. Straka, and G. Reitmayr. „Virtual Try-On through Image-Based Rendering“. In: *Visualization and Computer Graphics, IEEE Transactions on* 19.9 (2013), pp. 1552–1565.

[HSS+09] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. „A Statistical Model of Human Pose and Body Shape“. In: *Computer Graphics Forum* 28.2 (2009), pp. 337–346. ISSN: 1467-8659.

[HSW98] Jeffrey Huang, Xuhui Shao, and Harry Wechsler. „Face pose discrimination using support vector machines (SVM)“. In: *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*. Vol. 1. IEEE. 1998, pp. 154–156.

[Ike81] K. Ikeuchi. „Determining surface orientations of specular surfaces by using the photometric stereo method“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1981), pp. 661–669.

[JR02] Michael J. Jones and James M. Rehg. „Statistical color models with application to skin detection“. In: *Int. J. Comput. Vision* 46 (1 Jan. 2002), pp. 81–96. ISSN: 0920-5691.

[JSMH12] Eakta Jain, Yaser Sheikh, Moshe Mahler, and Jessica Hodgins. „Three-dimensional proxies for hand-drawn characters“. In: *ACM Trans. Graph.* 31.1 (Feb. 2012), 8:1–8:16. ISSN: 0730-0301.

[JTST10] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. „MovieReshape: Tracking and Reshaping of Humans in Videos“. In: *ACM Trans. Graph.* 29.5 (2010).

[KF08] Jiyeon Kim and Sandra Forsythe. „Adoption of Virtual Try-on technology for online apparel shopping“. In: *Journal of Interactive Marketing* 22.2 (2008), pp. 45–59.

[KJM08] Jonathan M Kaldor, Doug L James, and Steve Marschner. „Simulating knitted cloth at the yarn level“. In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM. 2008, p. 65.

[Lan12] M. Landgrebe. „Underworld: Awakening“. In: *Digital Production* 3 (2012).

[LC85] Hsi-Jian Lee and Zen Chen. „Determination of 3D human body postures from a single view“. In: *Computer Vision, Graphics, and Image Processing* 30.2 (1985), pp. 148 –168. ISSN: 0734-189X.

[LH95] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Ed. by C. L. Lawson and R. J. Hanson. Society for Ind. and Appl. Math., 1995, pp. 1–337. ISBN: 978-0898713565.

[LLN+10] Christian Lipski, Christian Linz, Thomas Neumann, Markus Wacker, and Marcus Magnor. „High Resolution Image Correspondences for Video Post-Production“. In: *Proc. European Conference on Visual Media Production (CVMP) 2010*. Vol. 7. 2010, pp. 33–39.

[LMG12] Olivier Le Meur and Christine Guillemot. „Super-resolution-based inpainting“. In: *Computer Vision–ECCV 2012*. Springer, 2012, pp. 554–567.

[LSS05] Bastian Leibe, Edgar Seemann, and Bernt Schiele. „Pedestrian detection in crowded scenes“. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 878–885.

[Mac+67] James MacQueen et al. „Some methods for classification and analysis of multivariate observations“. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[Mak] MakeHuman. *Make Human - Open Source tool for making 3D characters*. Website. `http://www.makehuman.org`, visited in April 2013.

[MBLS01] J. Malik, S. Belongie, T. Leung, and J. Shi. „Contour and texture analysis for image segmentation“. In: *International Journal of Computer Vision* 43.1 (2001), pp. 7–27. ISSN: 0920-5691.

[MM02] Greg Mori and Jitendra Malik. „Estimating Human Body Configurations Using Shape Context Matching“. In: *Computer Vision - ECCV 2002*. Ed. by Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen. Vol. 2352. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, pp. 150–180.

[MSMP08] Benjamin Meyer, Timo Stich, Marcus Magnor, and Marc Pollefeys. „Subframe Temporal Alignment of Non-Stationary Cameras“. In: *Proc. British Machine Vision Conference (BMVC) 2008*. Sept. 2008.

[NM65] J. A. Nelder and R. Mead. „A Simplex Method for Function Minimization“. In: *The Computer Journal* 7.4 (1965), pp. 308–313.

[PBBDN09] Marco Pasch, Nadia Bianchi-Berthouze, Betsy van Dijk, and Anton Nijholt. „Movement-based sports video games: Investigating motivation and gaming experience“. In: *Entertainment Computing* 1.2 (2009), pp. 49–61.

[PG04] Matt Pharr and Simon Green. „Ambient occlusion“. In: *GPU Gems* 1 (2004), pp. 279–292.

[PGB03] Patrick Pérez, Michel Gangnet, and Andrew Blake. „Poisson image editing“. In: *ACM Trans. Graph.* 22.3 (2003), pp. 313–318.

[PH03] David Pritchard and Wolfgang Heidrich. „Cloth Motion Capture.“ In: *Comput. Graph. Forum* 22 (2003), pp. 263–272.

[Pho75] Bui Tuong Phong. „Illumination for computer generated pictures“. In: *Commun. ACM* 18.6 (1975), pp. 311–317.

[PZB+09] Tiberiu Popa, Quan Zhou, Derek Bradley, Vladislav Kraevoy, Hongbo Fu, Alla Sheffer, and Wolfgang Heidrich. „Wrinkling Captured Garments Using Space-Time Data-Driven Deformation“. In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library. 2009, pp. 427–435.

[RBM14] Lorenz Rogge, Pablo Bauszat, and Marcus Magnor. „Monocular Albedo Reconstruction“. In: *Proc. IEEE International Conference*

on Image Processing (ICIP) 2014. IEEE, Oct. 2014, pp. 1046–1050.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. „Grabcut: Interactive foreground extraction using iterated graph cuts“. In: *ACM Transactions on Graphics (TOG)*. Vol. 23. 3. ACM. 2004, pp. 309–314.

[RKS+14] Lorenz Rogge, Felix Klose, Michael Stengel, Martin Eisemann, and Marcus Magnor. „Garment Replacement in Monocular Video Sequences“. In: *ACM Transactions on Graphics* 34.1 (Nov. 2014), 6:1–6:10.

[RKS12] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. „Reconstructing 3D Human Pose from 2D Image Landmarks“. In: *European conference on Computer Vision - Part IV*. Vol. 7575. Lecture Notes in Computer Science. 2012, pp. 573–586.

[RNWM11] Lorenz Rogge, Thomas Neumann, Markus Wacker, and Marcus Magnor. „Monocular Pose Reconstruction for an Augmented Reality Clothing System“. In: *Vision, Modeling and Visualization*. 2011, pp. 339–346.

[Rob94] L. Robert. „Camera calibration without feature extraction“. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing*. Vol. 1. 1994, pp. 704–706.

[RVHT12] M. Richter, K. Varanasi, N. Hasler, and C. Theobalt. „Real-time reshaping of humans“. In: *International Conference on 3D Imaging, Data Processing, Visualization and Transmission (3DimPVT)* (2012).

[SBB07] L. Sigal, A. Balan, and M. J. Black. „Combined discriminative and generative articulated pose and non-rigid shape estimation“. In: *Advances in neural information processing systems* (2007).

[SBB10] Leonid Sigal, AlexandruO. Balan, and MichaelJ. Black. „HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion“. English. In: *International Journal of Computer Vision* 87 (1-2 2010), pp. 4–27. ISSN: 0920-5691.

[SCF09] C. Scharfenberger, S. Chakraborty, and G. Farber. „Robust image processing for an omnidirectional camera-based smart car door“. In: *Embedded Systems for Real-Time Multimedia, 2009. ESTIMedia 2009. IEEE/ACM/IFIP 7th Workshop on*. Oct. 2009, pp. 106 –115.

[SF68] Irwin Sobel and Gary Feldman. „A 3x3 isotropic gradient operator for image processing“. In: (1968).

[SFPL10] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. „A state of the art in structured light patterns for surface profilometry“. In: *Pattern recognition* 43.8 (2010), pp. 2666–2680.

[SGA+10] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. „Video-based reconstruction of animatable human characters“. In: *ACM Trans. Graph.* 29.6 (Dec. 2010), 139:1–139:10. ISSN: 0730-0301.

[SH07] Jonathan Starck and Adrian Hilton. „Surface Capture for Performance-Based Animation“. In: *IEEE Computer Graphics and Applications* 27.3 (2007), pp. 21 –31.

[SHG+11] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt. „Fast articulated motion tracking using a sums of Gaussians body model“. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. Nov. 2011, pp. 951 –958.

[SM06] Volker Scholz and Marcus Magnor. „Texture Replacement of Garments in Monocular Video Sequences“. In: *Proc. Eurographics Symposium on Rendering (EGSR) 2006*. June 2006, pp. 305–312.

[SMFL00] Dimitrios Samaras, Dimitris Metaxas, Pascal Fua, and Yvan G Leclerc. „Variable albedo surface reconstruction from stereo and shape from shading“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE, 2000, pp. 480–487.

[SP06] Richard Swinbank and R. James Purser. „Fibonacci grids: A novel approach to global modelling“. In: *Quarterly Journal of the Royal Meteorological Society* 132.619 (2006), pp. 1769–1793. ISSN: 1477-870X.

[SSK+05] Volker Scholz, Timo Stich, Michael Keckeisen, Markus Wacker, and Marcus Magnor. „Garment Motion Capture Using Color-Coded Patterns“. In: *Computer Graphics Forum* 24.3 (2005), pp. 439–447. ISSN: 1467-8659.

[SSS06] Noah Snavely, Steven M. Seitz, and Richard Szeliski. „Photo tourism: exploring photo collections in 3D“. In: *ACM Trans. Graph.* 25.3 (2006), pp. 835–846.

[TAL+07] Christian Theobalt, Naveed Ahmed, Hendrik Lensch, Marcus Magnor, and Hans-Peter Seidel. „Seeing People in Different Light-Joint Shape, Motion, and Reflectance Capture“. In: *IEEE Transactions on Visualization and Computer Graphics* 13.4 (2007), pp. 663–674.

[TE12] Daniel Thiele and Rolf Ernst. „Optimizing performance analysis for synchronous dataflow graphs with shared resources“. In: *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium. 2012, pp. 635–640.

[Thi] J. Thingvold. *Biovision BVH format*. `http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html`.

[Tip00] Michael E. Tipping. „The relevance vector machine“. In: *Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 652–658.

[TOCR11] Eno Töppe, MartinR. Oswald, Daniel Cremers, and Carsten Rother. „Silhouette-Based Variational Methods for Single View Reconstruction“. In: *Video Processing and Computational Video.*

Ed. by Daniel Cremers, Marcus Magnor, MartinR. Oswald, and Lihi Zelnik-Manor. Vol. 7082. Lecture Notes in Computer Science. 2011, pp. 104–123. ISBN: 978-3-642-24869-6.

[VC02] L.A. Vese and T.F. Chan. „A multiphase level set framework for image segmentation using the Mumford and Shah model“. In: *International Journal of Computer Vision* 50.3 (2002), pp. 271–293. ISSN: 0920-5691.

[VJ01] Paul Viola and Michael Jones. „Robust real-time object detection“. In: *International Journal of Computer Vision* 4 (2001), pp. 34–47.

[VN08] V. Vineet and P. J. Narayanan. „CUDA cuts: Fast graph cuts on the GPU“. In: *Computer Vision and Pattern Recognition Workshop* 0 (2008), pp. 1–8.

[VPB+09] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. „Dynamic shape capture using multi-view photometric stereo“. In: *ACM Transactions on Graphics (TOG)* 28.5 (2009), p. 174.

[VSHJ12] Marek Vondrak, Leonid Sigal, Jessica K. Hodgins, and Odest Chadwicke Jenkins. „Video-based 3D motion capture through biped control“. In: *ACM Trans. Graph.* 31.4 (2012), p. 27.

[VWB+12] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. „Lightweight binocular facial performance capture under uncontrolled lighting.“ In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), p. 187.

[WC10] Xiaolin Wei and Jinxiang Chai. „VideoMocap: modeling physically realistic human motion from monocular video sequences“. In: *ACM Trans. Graph.* 29.4 (2010), 42:1–42:10.

[WKK+04] Markus Wacker, Michael Keckeisen, Stefan Kimmerle, Wolfgang Straßer, Volker Luckas, Clemens Groß, Arnulph Fuhrmann, Mirko Sattler, Ralf Sarlette, and Reinhard Klein. „Virtual try-on“. In: *Informatik-Spektrum* 27.6 (2004), pp. 504–511.

[Wol13] Yannick Wolff. *Parkour*. http://vimeo.com/68317895. 2013.

[Woo79] Robert J Woodham. „Photometric stereo: A reflectance map technique for determining surface orientation from image intensity“. In: *22nd Annual Technical Symposium*. International Society for Optics and Photonics. 1979, pp. 136–143.

[Woo80] Robert J. Woodham. „Photometric Method For Determining Surface Orientation From Multiple Images“. In: *Optical Engineering* 19.1 (1980), pp. 139–144.

[WP09] Robert Y. Wang and Jovan Popović. „Real-time hand-tracking with a color glove“. In: *ACM Trans. Graph.* 28 (3 July 2009), 63:1–63:8. ISSN: 0730-0301.

[WVL+11] Chenglei Wu, Kiran Varanasi, Yebin Liu, H-P Seidel, and Christian Theobalt. „Shading-based dynamic shape refinement from multi-view video under general illumination“. In: *IEEE International Conference on Computer Vision*. IEEE. 2011, pp. 1108–1115.

[XLS+11] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. „Video-based characters: creating new human performances from a multi-view video database“. In: *ACM Trans. Graph.* 30.4 (2011), 32:1–32:10.

[YD12] Wentao Yao and Zhidong Deng. „A robust pedestrian detection approach based on shapelet feature and Haar detector ensembles“. In: *Tsinghua Science and Technology* 17.1 (2012), pp. 40–50.

[YKJM12] Cem Yuksel, Jonathan M Kaldor, Doug L James, and Steve Marschner. „Stitch meshes for modeling knitted clothing with yarn-level detail“. In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), p. 37.

[YLK11] Jong-Chul Yoon, In-Kwon Lee, and Henry Kang. „Image-based dress-up system“. In: *Ubiquitous Information Management and Communication*. ICUIMC '11. 2011, 52:1–52:9.

[ZC91] Qinfen Zheng and Rama Chellappa. „Estimation of illuminant direction, albedo, and shape from shading“. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1991, pp. 540–545.

[ZCW10] Nan Zhang, Yun-shan Chen, and Jian-Li Wang. „Image parallel processing based on GPU“. In: *Advanced Computer Control (ICACC), 2010 2nd International Conference on*. Vol. 3. IEEE. 2010, pp. 367–370.

[Zel05] Cyril Zeller. „Cloth simulation on the GPU“. In: *ACM SIGGRAPH 2005 Sketches*. ACM. 2005, p. 39.

[ZFL+10] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. „Parametric reshaping of human bodies in images“. In: *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, 2010, 126:1–126:10. ISBN: 978-1-4503-0210-4.

[ZLA+04] G. Ziegler, H.P.A. Lensch, N. Ahmed, M. Magnor, and H.-P. Seidel. „Multivideo compression in texture space“. In: *International Conference on Image Processing*. Vol. 4. 2004, 2467 –2470 Vol. 4.

[Goo] Google, Inc. *Google Glass*. Website. `https://www.google.com/glass`, visited in November 2014.

[How11] Howcast Media, Inc. *Ballet Dancing: How to Do a Pirouette*. http://www.howcast.com/videos/497190-How-to-Do-a-Pirouette-Ballet-Dance. 2011.

[Lay] Layar, Inc. *Layar*. Website. `https://www.layar.com`, visited in March 2015.

[MGM05] MGM, Inc. *Into the Blue*. http://www.mgm.com/#/our-titles/969/Into-the-Blue. 2005.

[Meta] Metaio GmbH. *Metaio - The Augmented Reality Company*. Website. `http://www.metaio.com`, visited in November 2014.

[Metb] Metaio GmbH. *The IKEA Catalogue app*. Website. `http://www.ikea.com/gb/en/catalogue-2015/`, visited in May 2015.

[Org] Organic Motion, Inc. *Stage Markerless Motion Capture System*. Website. `http://www.organicmotion.com/solutions/stage`, visited in July 2011.

[Son] Sony Computer Entertainment, Inc. *Eye Of Judgment*. Website. `http://en.wikipedia.org/wiki/The_Eye_of_Judgment`, visited in November 2014.

[Sto11] Stock Footage, Inc. *Man doing yoga on a white background*. http://www.stockfootage.com/shop/man-doing-yoga-on-a-white-background. 2011.

[The] The Foundry, Inc. *Nuke - Powerful node based VFX, editorial & finishing tools*. Website. `http://www.thefoundry.co.uk/products/nuke`, visited in March 2015.